2019

Synergistic Visualization And Quantitative Analysis Of Volumetric Medical Images

Neslisah Torosdagli
University of Central Florida

SYNERGISTIC VISUALIZATION AND QUANTITATIVE ANALYSIS OF VOLUMETRIC MEDICAL IMAGES

by

NESLİŞAH TOROSDAĞLI
M.S., University of Central Florida, 2016

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2019

Major Professor: Ulaş Bağcı

ABSTRACT

The medical diagnosis process starts with an interview with the patient, and continues with the physical exam. In practice, the medical professional may require additional screenings to precisely diagnose. Medical imaging is one of the most frequently used non-invasive screening methods to acquire insight of human body. Medical imaging is not only essential for accurate diagnosis, but also it can enable early prevention. Medical data visualization refers to projecting the medical data into a human understandable format at mediums such as $2D$ or *head-mounted* displays without causing any interpretation which may lead to clinical intervention. In contrast to the medical visualization, quantification refers to extracting the information in the medical scan to enable the clinicians to make fast and accurate decisions.

Despite the extraordinary process both in medical visualization and quantitative radiology, efforts to improve these two complementary fields are often performed independently and synergistic combination is under-studied. Existing image-based software platforms mostly fail to be used in routine clinics due to lack of a unified strategy that guides clinicians both visually and quantitatively. Hence, there is an urgent need for a bridge connecting the medical visualization and automatic quantification algorithms in the same software platform. In this thesis, we aim to fill this research gap by visualizing medical images interactively from anywhere, and performing a fast, accurate and fully-automatic quantification of the medical imaging data. To end this, we propose several innovative and novel methods. Specifically, we solve the following sub-problems of the ultimate goal: (1) direct web-based out-of-core volume rendering, (2) robust, accurate, and efficient learning based algorithms to segment highly pathological medical data, (3) automatic landmarking for aiding diagnosis and surgical planning and (4) novel artificial intelligence algorithms to determine the sufficient and necessary data to derive large-scale problems.

To the memory of my beloved cousin, Hakan Tuntaş

ACKNOWLEDGMENTS

This journey would not have been possible without the support of my professors, my friends and my family.

I would like to express my deepest appreciation to my doctoral advisors Dr. Sumanta Pattanaik and Dr. Ulaş Bağcı for their excellent and continuous guidance, caring, patience, and providing me with an outstanding environment for doing research. Without their support, it would be impossible for me to accomplish.

My sincere gratitude to Dr. Curtis Lisle for his comprehensive knowledge, positive attitude and guidance.

My kind thanks to Dr. Yanling Liu for funding my PhD through Frederick National Laboratory for Cancer Research for 4 semesters.

My genuine thanks to the researchers at the National Institute of Dental and Craniofacial Research (NIDCR) for sharing their knowledge and for their endless support.

My deepest thanks to my friends Awrad, Neda, Sahar, and AliAsghar, Harish, Naji, Rodney, Samira and Sarfaraz for their support and friendship.

My sincere gratitude to the Balcı family, Derya, Derin, Ege and Murat for encouraging me to start my PhD journey and for their everlasting support.

My heartfelt gratitude to my mother Sabuş, my sister Gülşah and my niece Derin for their enduring support during my sleepless and stressful PhD journey, and especially for taking care of us during this period. Last but not the least, I would like to express my heartful thanks to my sons Doğan and Kaan, and to my husband Tunç for their patience, motivation and love!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The ultimate goal of the clinicians is to provide correct and fast clinical interpretation of the diseases and provide treatment options. After the clinical examination, medical imaging is one of the most frequently used tools by the clinicians for diagnosis, prognosis, therapy planning, and management of the diseases. There are a wide range of imaging modalities used to acquire the medical data such as X-ray, computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, and positron emission tomography (PET). Each modality is well suited to evaluate different types of the tissues and different types of the anomalies. 2D X-ray, which is the cheapest, most typical and most-widely used modality all around the world is introduced by Wilhelm Röntgen in 1895. In 1968, Godfrey Hounsfield and Allan McLeod introduced the first volumetric scanner - CT, which was recognized by a Nobel prize in 1979. Investigation of CT opened a new era in the medical visualization world. MRI is introduced in 1973 by Lauterbur and Mansfield. CT employes X-rays to acquire medical data, in contrast, MRI utilizes radio waves and powerful magnetic fields. CT scans are more often used for exploring the bony regions and the tumors, while MRI is used for exploring mainly the soft tissues. Compared to CT and MRI, the ultrasound scanners are based on the high frequency sound waves. Although, precise reading of the ultrasound requires considerable experience, it is a cheap real-time imaging modality. Nuclear imaging is the method used to explore the metabolic activity. Nuclear imaging such as PET is usually used as hybrid PET/CT or PET/MRI scans [5].

1.1 Medical Data Visualization

In the old days, an X-ray light box was sufficient to visualize and explore the information of a $2D$ X-ray. Volumetric scan is a series of $2D$ scans. If there are no visualization tools available, clinicians still employ an X-ray light box to visually explore the small scaled $2D$ slices one by one. However, this is a very tedious process, and it is more likely to miss the useful $3D$ volumetric information available. The invaluable insight of the human body provided by the medical scans is useful to the extent it is interpreted. To visualize, volumetric medical data is projected to the viewing space in such a way that the mapping is correct, and visually meaningful and understandable frames are generated at interactive rates. There are multiple number of medical data visualization tools such as Slicer3D [6]. However, to our knowledge none of the tools available serves to the current computer technology where the cloud-based mobile devices are trending.

1.2 Quantitative Image Analysis

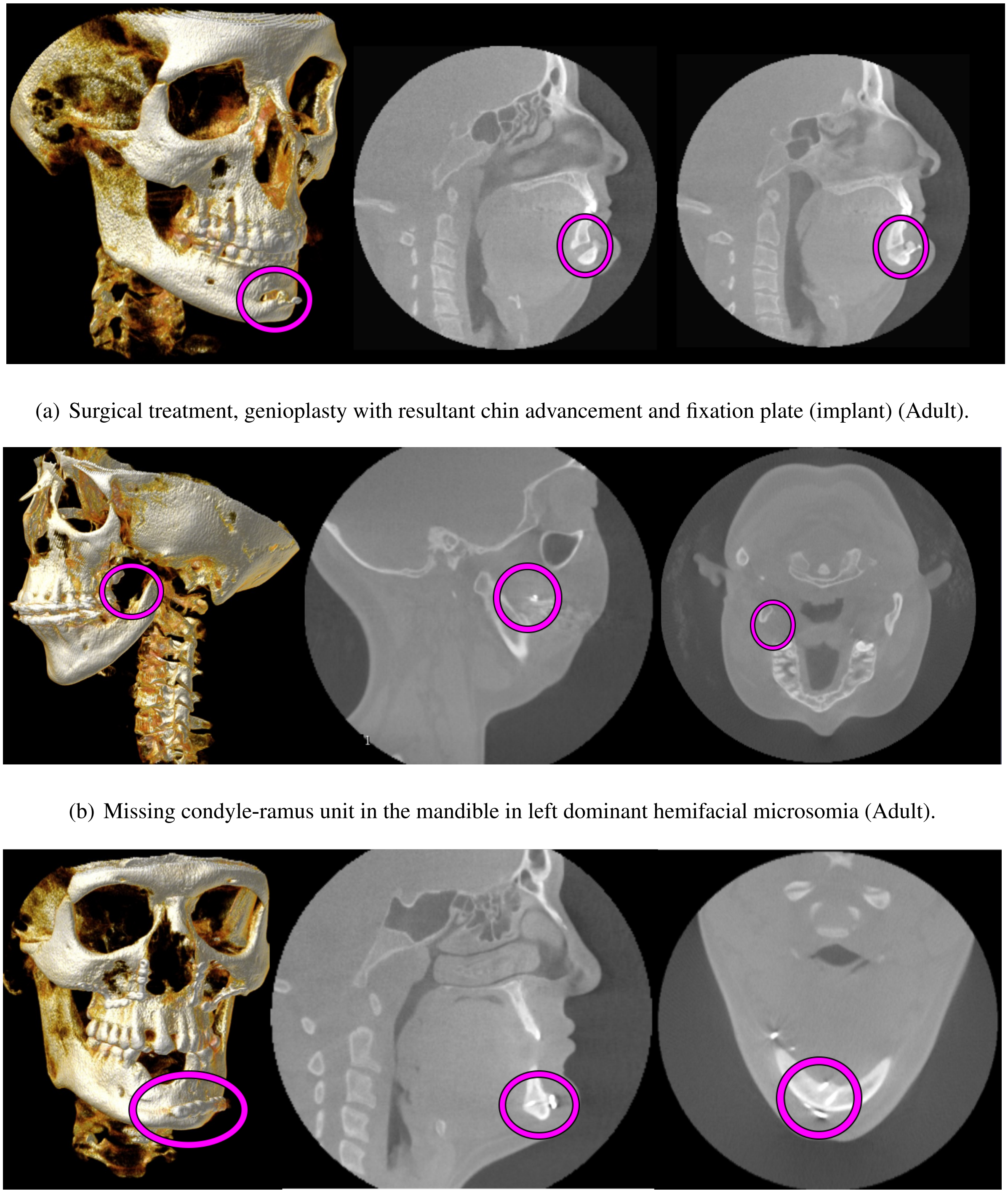
The core image analysis framework in the bony regions is based on the variations of the CT scans. This framework includes detection and measurement of the abnormality, which requires precise segmentation of the bones in the region of interest (ROI). However, automatically and accurately segmenting bones is a significant challenge and still an open problem. The statement "Fully automatic, completely reliable segmentation in medical images is an unrealistic expectation with today's technology"[7] is still true when especially low-quality scans taken in the hospitals and serious deformities are considered [5]. Measurement of abnormalities is performed through anatomically distinct locations (called landmarks). Yet another challenge and open problem introduced is automatic and accurate localization of the landmarks in the ROI. Severity of the diseases, as well as the treatment/surgery plans are built upon these measurements.

1.3 Clinical Motivations of this thesis

Craniomaxillofacial (CMF) is the head region which consists of all the structures of the mouth, the jaws, the face, and the skull. There are more than 17 million patients with the congenital and the developmental deformities in the craniomaxillofacial (CMF) region in the United States. Post-traumatic defects, deformities of the temporamandibular joints, defects of tumor ablation, and congenital diseases are some of the leading causes of these deformities [8]. The number of patients who need orthodontic treatment is far beyond this number. Among the CMF conditions, deformation or injury in the mandibular region is the most prevalent condition. For instance, a significant percent of the skeletal traumas affects the facial area, of which around 76% is in the mandibular region [9].

Cone-beam computed tomography (CBCT) scan, a variation of traditional computed tomography (CT), is the conventional imaging modality for the diagnosis and treatment planning of the patients with CMF deformities. It is because not only the CBCT scanners produces higher resolution data and lower radiation dose compared to the traditional CT scanners, but also CBCT scanners are compact, fast and less expensive, which makes them affordable in the office settings.

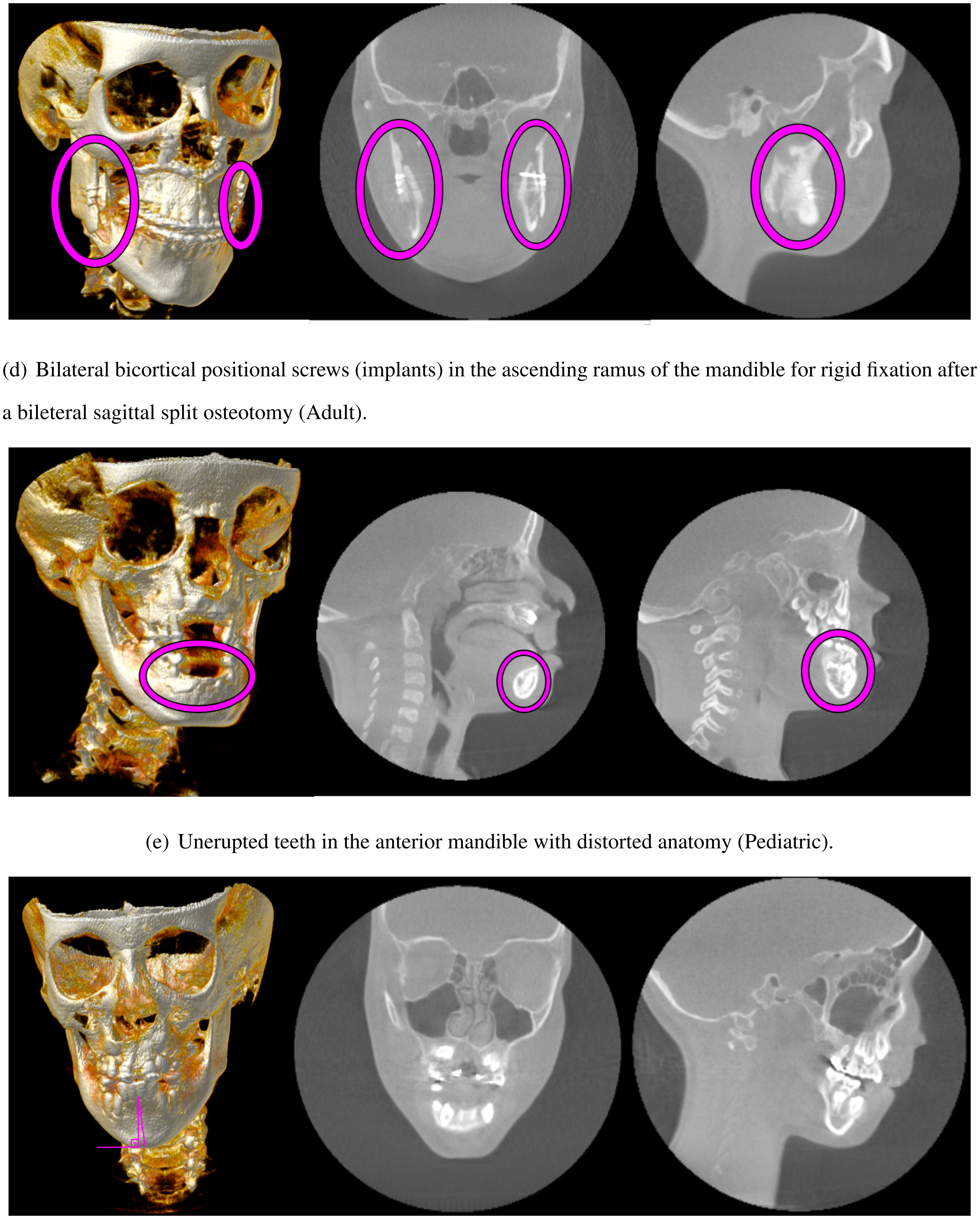
CBCT based image analysis framework, specifically segmentation of CMF bones and measurements of certain anatomical locations in CMF regions, plays a significant role in diagnosing the diseases, understanding severity, planning the treatment options, and estimating the risk of potential interventions. However, automatically segmenting the bones from CMF regions and identifying the clinically justified anatomical landmarks in these bones for deformation analysis is a significant challenge and still an open problem. Despite some recent elaborative efforts towards making a fully automated and accurate software for segmentation and landmark digitization for dental applications, the problem remains unsolved for general purpose CMF deformity analysis.



(a) Surgical treatment, genioplasty with resultant chin advancement and fixation plate (implant) (Adult).

(b) Missing condyle-ramus unit in the mandible in left dominant hemifacial microsomia (Adult).

(c) Plate and screws (implants) in the anterior mandible for rigid fixation and reduction of an oblique fracture (Adult).



(d) Bilateral bicortical positional screws (implants) in the ascending ramus of the mandible for rigid fixation after a bileteral sagittal split osteotomy (Adult).

(e) Unerupted teeth in the anterior mandible with distorted anatomy (Pediatric).

(f) Mid-sagittal plane with respect to lower jaw incisors have a serious degradation from the 90 degrees (Pediatric).

Figure 1.1: Some examples in our dataset.

The main reason for this research gap is the challenging nature of the problem due to an extremely high variation in the anatomy of the bones and their deformities in the CMF regions. Figure 1.1 shows some of the known deformities in the CMF, including the missing bones, and the surgical effects. Other reasons are due to the imaging artifacts of the CBCT such as noise, inhomogeneity, truncation, beam hardening, and low resolution. Due to all these problems, clinicians still perform their analysis either manually or semi-automatically with a limited software support. However, manually extracting the quantitative information from a $3D$ CBCT scan is an extremely tedious process and prone to the repeatability problems. Eventually, there is a strong need for creating a general purpose CMF image analysis platform that can help clinicians (1) to visualize, (2) to create a surface model (through segmentation) and (3) to define anatomically known points (called landmarks) for precise deformation analysis. Hence, this thesis is focused on:

- Development of web-based volumetric medical data visualization framework (Chapter 3),
- Development of fast and robust CMF bone segmentation algorithms (Chapters 4 and 5),
- Development of learning-based CMF landmark localization algorithms (Chapters 5 and 6)

The rest of this chapter will explain more and necessary details on Medical Data Visualization to make this thesis self-explanatory. Literature survey will be provided in Chapter 2. Chapters 3, 4, 5 and 6 are concerned with the methodologies employed. Finally, Chapter 7 will conclude the dissertation .

1.4 Medical Visualization

Medical scan is a volumetric data which is simply a $3D$ array of intensity values (voxels). The intensity values usually range from $-2000s$ to $3000s$ in the CBCT scans.

Volumetric visualization consists of a set of techniques to extract and to display a $2D$ graphical representation of the meaningful information in the volumetric data. The visualization can be performed using (1) Indirect or (2) Direct approaches.

In the **indirect volume visualization** approach, in a pre-processing step, volumetric data is mapped into a surface (mesh) model which is composed of graphical primitives. After generation of the mesh model, traditional Computer Graphics rendering techniques are applied. Marching cubes is a common Indirect Volume Rendering algorithm [10, 1]. This algorithm uses a threshold based segmentation technique. Using a user set threshold, iso-value, each voxel is classified as inside the object, outside the object or on the boundary of the object. Triangulation follows the classification step, it is computed by visiting the volume grid cells one at a time. The classification status of the corners of the grid cell determines if the threshold surface intersects the grid cell and if does then determines the polygonal shape of the intersecting surface. The final mesh is constructed by combining all such polygonal shapes.

The main disadvantage of the marching cubes algorithms is that the mesh model generated may be complex and may contain holes. In addition, multiple-object mesh generation requires multi-pass application of the algorithm.

In contrast, in the **direct volume rendering (DVR)** approach, useful information in the volumetric data is interactively extracted according to the camera position and lighting status of the scene. The rendering is performed without explicitly generating the surface model. In this approach, each voxel is assumed to be a light emitting, reflecting, scattering, absorbing and occluding entity [11]. Light passing through the volume is simulated and final rendering is obtained by computing the interaction of each voxel with the light [12] in the field of view.

Direct volume rendering is a well-studied field-of-study. There are open-source desktop applications with DVR capability such as 3D Slicer [6], which can be freely downloaded and used.

However, the current trend in computing is towards the on-line applications which can be accessed from anywhere including a wide range of computing platforms. Eventually, web based applications are becoming increasingly popular because browsers are supported on nearly all computing platforms, and the new HTML5 standards enable graphical interaction without any need for proprietary plugins or additional APIs.

The size of modern day biological/medical volumes increases by the advancements in the volumetric capture devices. Volume rendering algorithms are easily parallelizable and make effective utilization of the GPU processing capabilities using OpenCL/Cuda or OpenGL/DirectX APIs [13]. However, the available GPU memory (and even CPU memory) is often inadequate for the size of modern day biological/medical volumes. Ultimately, volume rendering is still challenging for larger volumes that exceed the resources available on the rendering hardware. This is particularly true for the mobile clients. Although there are some large-scale volumetric visualization tools available such as Vaa3D-TeraFly [14, 15, 16], none of them fulfill requirements especially for the mobile clients. Hence comes the need for a volume visualization architecture that can scale in performance by using server-based computing, yet deliver the volumes to small or mobile devices.

In this thesis, **out-of-core techniques** have been proposed to handle the above mentioned problem [16, 17, 18]. Such techniques often use a standard memory-paging scheme, in which large volumes are broken into smaller chunks (called bricks) such that multiple bricks can easily fit into the GPU memory.

The wide availability of the GPU-cloud based systems enable development of the web-based out-of-core volume rendering platforms using a client-server architecture. The compute/storage/memory intensive component is executed on a server equipped with high-performance CPU and/or GPU hardware. The output is then delivered to one or more (usually less powerful) client devices such as smart-phones through a connected network.

The literature review for (1) Web-based volumetric data visualization, (2) CMF bone segmentation (3) CMF bone landmark localization is presented in the following Chapter 2.

CHAPTER 2: LITERATURE REVIEW

2.1 Volumetric Medical Data Visualization

The term *volumetric medical data* has been used to describe $3D$ medical data, hence *volumetric medical data visualization* refers to $3D$ medical data visualization. By the advent of the modern medical visualization technologies, “invisible depths” are turned into “visible surfaces”[19]. Traditional $2D$ X-ray visualization means are not sufficient to visualize this invaluable $3D$ insight into the invisible parts of the body, but rather volumetric medical data visualization algorithms are needed. Volumetric medical data visualization algorithms should allow projecting the meaningful information in the volumetric medical dataset to the $2D$ viewing screen in such a way that $3D$ to $2D$ mapping is correct and meaningful, and understandable frames are generated interactively [20, 21].

There are two typical volumetric data visualization approaches Indirect (aka. surface rendering) and Direct Volume Rendering.

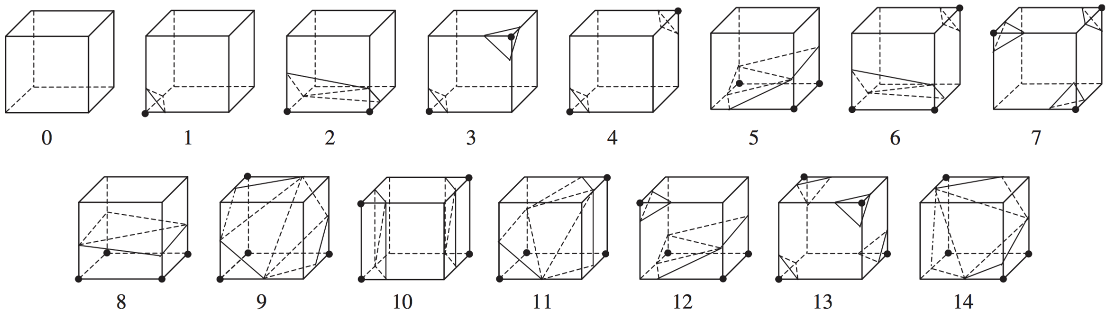


Figure 2.1: Marching Cubes algorithm: possible marking scenarios and facetization when reflective and rotational symmetries are considered [1].

2.1.1 Indirect Volume Rendering

The existing literature on indirect volume rendering is extensive and focuses particularly on *iso-surface* algorithms, root of which is the marching cubes algorithm [10]. Marching cubes algorithm [1] is successfully applied in a wide range of application areas including biochemistry [22], biomedicine [23], and environmental sciences [24], etc. Eventually, many variations of the standard algorithm [10] are proposed.

The standard algorithm, in the first pass, assigns each voxel to an iso-value or leaves it unmarked. Black dots on the corners of the cubes in Figure 2.1 are the iso-values. In the second pass, facets, dashed line in Figure 2.1, are generated. When no symmetry is considered, for each cube there are $2^8(256)$ different iso-value markings possible. Taking the reflective and rotational symmetries into consideration, the number of possible markings is decreased to 15 (Figure 2.1). The intersection points of the facets are estimated using linear interpolation. The triangular primitives are computed based on the facets generated in the previous step.

At the end of 1990, Payne et al. [25] applied marching cubes algorithm to visualize the cortical surface, and the curvilinear surfaces deep in the cortex with the aim to achieve both realistic and quantitatively precise $2D$ renderings. Similarly, in 1991, Wallin et al. applied the marching cubes algorithm to visualize CT scans to display the normal and also the pathological anatomies [26]. In 1993, Takanori et al. [27] extended the marching cubes algorithm to resolve the ambiguous cases and verified the extended algorithm on a CT dataset.

In 2001, Delibasis et al. [28] proposed a generic rule-based approach where the ambiguities such as 'hole problem' are resolved, and applied to full resolution CT and MRI dataset of human heads. Meanwhile, Tory et al [29] visualized time-varying data in two different modalities: MRI and dynamic SPECT. The authors applied both iso-surfaces and direct volume rendering approaches, and compared their performances. In 2003, Yim [30] proposed a deformable iso-surface algorithm and applied to the magnetic resonance angiography (MRA) of three renal arteries with anomalies.

Several other works in the literature focused on the number of the marking scenarios. In [31, 32, 33] 14 scenarios rather than 15 are proposed. Later, the algorithm is extended to serve for variations of data such as multi-resolution data [34] and higher dimensional data [35]. There are also efforts to extend the traversal methods to octree traversal [36]. For a detailed discussion on the variations of the marching cubes algorithm, the survey [1] is recommended.

Limitations: iso-surface algorithms is that multiple-object generation requires multi-pass execution of the algorithm. In [37] Gerstner proposed a method to interactively generate multiple transparent iso-surfaces using the tetrahedral bisection hierarchy. Compared to direct volume rendering approaches, the computational complexity of the proposed method is simple. However, the authors applied the algorithm to comparably small-sized scans. The proposed algorithm also suffers from losing the details between the iso-surfaces.

2.1.2 Direct Volume Rendering (DVR)

In the direct volume rendering (DVR) approach, useful information in the volumetric data is interactively extracted according to the camera position and lighting status of the scene without creating any models. The first direct volume rendering algorithms are proposed simultaneously in 1988 by Dreblin et al. [38] and by Levoy [39] and the current state-of-art direct volume rendering algorithms are based on these 2 studies. The rendering pipeline proposed by Levoy [39] starts with data preparation (pre-processing), where each voxel was assigned a shaded color and opacity in the parallel shading and the classification steps respectively. Blinn-Phong shading [40, 41] was implemented in the shading step where normals are approximated with the input data gradients. In the last step, sampled color and opacity values were composited using linear interpolations [39, 42].

This algorithm, which is also known as ray casting, requires just two optical properties: the color and the opacity. Similar to modern approaches using lookup-table transfer functions, Levoy [39] also mentioned using a $2D$ lookup table. However, the intention of this lookup table was just the opacity computation.

In the modern volume rendering approaches multi-dimensional transfer functions are employed for both color and opacity assignments. However, designing a transfer function, which is equivalent to accurate segmentation of the structures in the volumetric data is a challenging, and an open problem.

The core direct volume rendering algorithms after the pioneers [39, 38, 12] has come to a maturity point in the beginning of $2000s$. Further studies rather than improving the core algorithm focuses on visualizing the out-of-core data which do not fit into the GPU memory [17, 43], multi-core implementation of the algorithm [44, 45, 46], or semi-or-fully automatic design of the transfer functions [47, 48]. Designing a transfer function or multi-core implementation of the algorithm is out of scope of this dissertation. Literature survey of the out-of-core volume rendering algorithms is detailed in the next section.

2.1.3 Out of Core Volume Rendering

GPU accelerated volume rendering approaches are introduced in the beginnings of $2000s$ [49]. Despite the highly-parallel nature of the volume rendering algorithm and high performance gains of the GPU-based algorithms, the large gap between the limited GPU memory and big data sizes and the long GPU-to-CPU (and CPU-to-GPU) transfer durations introduced new challenges to the researchers.

In 2003 Hadwiger et al. [50] proposed a two-level GPU-based volume rendering approach using different rendering approaches and different transfer functions for each tissue on the segmented dataset. The tissues that share the same rendering mode are processed in the same pass. Application of alpha-blending and mip-map construction (MIP) yielded a high-quality GPU implementation. The proposed approach is verified on a comparably smaller $256 \times 128 \times 256$ dataset. Later in 2008 Gobbetti et al. [51] employed a hierarchical octree structure. In this octree structure, the leaves are the original resolution bricked data, whereas the root is the whole scan downsampled to the brick size. View and transfer function-dependent working set of the bricks are adaptively transferred to the GPU. The indexing structure spatially organizes the working set into an octree hierarchy. In 2009 Crassin et al. [17] traversed the $N^3 - tree$ data structure to achieve a high quality rendering at interactive rates with billions of voxels. In 2010 Fogal et al. [43] demonstrated that the inability to switch the data sampling rate and data sizes are the major bottlenecks of the volume rendering frameworks, and implemented a ray-guided volume rendering application. Later in 2011 Engel et al. [52] proposed a framework to interactively render teravoxel scale volume on standard PCs using a progressive multi-resolution out-of-core volume rendering approach. Data loading was initiated as needed by the rays that are cast through the volume. Eventually, occluded or empty data were never transferred to the GPU. Meanwhile, in 2011 Knoll et al [53] introduced a fast and scalable CPU volume rendering framework. The proposed approach did not perform well for smaller datasets, however its performance was comparable to the performance of machines with strong GPUs for larger datasets.

Limitations: Although brick-based approaches resolve the discord of data sizes and GPU memory limitations to a degree, access of the bricks at the GPU introduce additional latencies. We address this problem by proposing a different index structure in our volume rendering framework based on [51]. Furthermore, there has not been any documented study on web-based out-of-core medical data visualization. To address this, we proposed a web-based client-server volume rendering framework [54] which enables employing a scalable cloud-based server.

2.1.4 Web-based Medical Data Visualization

The first web browser was introduced by Tim Berners-Lee in 1990. In the beginning, the web pages have static text content and remote pages are shared on the internet. In spite of the 28 years of the web development, web-based visualization of medical data is started recently in the beginning of $2010s$. In practice, medical visualization algorithms are computationally complex and dedicated high-end machines are employed to visualize the medical data. In addition, medical data sizes increase rapidly each year. In 2010, Settapat et al. [55] proposed an iso-surface rendering application in a client-server architecture using the Web3D standard. Later in 2012 Jacinto et al. [56] proposed a client-server medical image processing framework utilizing WebGL technologies. A standalone WebGL based medical data visualization application [57] has been proposed in this thesis that allows employing customizable transfer functions.

2.1.5 Web-based Out-of-Core Medical Data Visualization

To our knowledge, there has not been any documented study on web-based out-of-core medical data visualization. The work [54] presented in this thesis fills this gap. The study is a volume rendering framework based on [51]. The proposed framework works in a Client-Server architecture enabling a scalable cloud-based server.

2.2 Medical Data Segmentation

Historically, active contour, shape, and appearance models have been used extensively to segment CMF bones [58]. In the later years, energy based segmentation methods such as Markov Random Field (MRF) and Conditional Random Fields (CRF) have replaced such methods. In parallel, registration (atlas-based) methods [59] have been reported to achieve relatively higher accuracy when shape and appearance information are integrated. However, registration methods suffer from the computational complexity where convergence may take several hours [4]. Furthermore, nonlinear registration may not be sufficient when the large morphological variances among the patients are considered. In contrast to the model and registration-based methods, learning based methods become popular recently. Basically, these methods employ either hand-crafted or self-learned features by utilizing machine learning classifiers. With the rise of the deep learning methods, the field of medical image segmentation shifted to depend largely on the learning based methods. These features are fed into the learning network to perform the segmentation. In 2015, Ronneberger et al. [60] employed a U-Net framework to perform biomedical image segmentation and this method has been considered as one of the state-of-art segmentation methods. Çiçek et al in 2016 extended this work by employing 3D convolutional kernels. Meanwhile, Milletari et al. [61] applied both 3D convolutional kernels and also residual blocks to segment prostate MRI scans. Later in 2017, Kayalibay et al. [62] applied an U-Net-like architecture with three-dimensional filters to segment hand and brain MRI scans. Examples are vast; hence we will only consider major innovations in this field.

2.2.1 CMF Bone Segmentation

In 2012, Gollmer et al. [63] applied a statistical shape model (SSM) to segment mandible from the CBCT scans. In 2017, Zhang et al. [4] and Chuang et al. [64] also proposed two separate studies to segment the mandible. Zhang et al. [4] employed a cascaded U-Net architecture to segment the head-neck CT scans into 2 groups as midface and mandible. While, Chuang et al. [64] developed a registration-based semiautomatic mandible segmentation (SAMS) framework. The initial work presented in this thesis [65] also proposed a data-driven random forest regression and fuzzy connectivity approach to segment Mandible in CT scans. In this thesis, we further improved our segmentation framework via deep-learning based algorithms, and employed a Fully

Convolutional DenseNet architecture by which we obtained state-of-art segmentation accuracies.

Despite all available research methods to perform medical data segmentation, to our knowledge there isn't any commercial software available to perform fast CMF bone segmentation accurately and automatically. Hence, in practice, the clinicians perform the CMF bone segmentation manually.

2.3 Anatomical Landmark Localization

There are two main approaches to localize landmarks on the medical images: 1) model-based and 2) machine-learning. In recent years, machine learning approaches have proven to be more successful compared to the atlas-based ones.

In 2008, Zhan et al. [66] employed confidence maximizing sequential scheduling to detect anatomical landmarks of multiple organs. In 2011, Cheng et al. [67] applied a random forest classifier to localize anatomical dental landmarks. Meanwhile, Zhan et al. [66] applied a cascaded Adaboost classifier to localize knee anatomical landmarks on the MRI scans. In 2013, Criminisi et al. published a nice study [68] to predict the $3D$ displacement of each voxel via regression forest. Likewise Cheng et al, Ebner et al. [69] in 2014 also employed multiple random regression forests to localize the joints between the hand bones. Meanwhile in 2014, Gao et al. [70] employed a two-layer context-aware regression forest to localize the prostate landmarks. Following Gao, Chen et al. [71, 72] proposed a data-driven approach to localize landmarks on X-ray and also on MR scans of intervertebral discs. In 2015, Cootes et al. [73] extended Criminisi et al. [68] method to detect facial anatomical landmarks. Similar to Ebner [69], Payer et al. [74] employed a fully convolutional network (FCN) for landmark heatmap regression to localize anatomical landmarks on two different datasets of hand images (2D radiographs, and 3D MRI). Although Payer et al used a limited dataset, promising experimental results are achieved.

2.3.1 Relational Reasoning

In 2009, Scarselli et al. [75] introduced graph neural network (GNN) by extending the neural network models to process graph data which encoded relationship information. Later in 2016 Li et al. [76] proposed a machine learning model based on the gated recurrent units to learn the distributed vector representations from heap graphs. Despite the promising nature of the GNN architectures [77], there is limited understanding of their representational properties and limitations.

Recently, four important works were published for relational reasoning of the objects by the DeepMind teams, Battaglia et al. [78], Raposo and Santoro et al. [79], Santoro and Raposo et al. [80] and Battaglia et al. [81]. Battaglia et al. [78] introduced interaction networks to reason about the objects and the relations in the complex environments. The authors proposed a simple and accurate system to reason about difficult problems such as n-body problems, rigid-body collision, and non-rigid dynamics. The proposed system can predict the dynamics in the next step with order of magnitude lower error and higher accuracy. Later in 2017, Raposa and Santoro et al. [79] introduced the Relational Network (RN) which can learn the object relations from the scene description. Following this study, Santoro and Raposa et al. [80] presented a relational reasoning architecture for tasks such as visual question-answer, text-based question-answer, and dynamic physical systems. The proposed model is highly accurate and gets most answers correctly. In 2018, Battaglia et al. [81] studied the relational inductive biases to learn the relations of the entities and presented the graph networks. To the best of our knowledge, such advanced reasoning algorithms have neither been developed for nor applied to the medical imaging applications. Besides, medical imaging applications require fundamentally different reasoning paradigms as the anatomy-anatomy and anatomy-pathology relationships are extremely different from the conventional object-object relationships in conventional vision tasks.

2.3.2 CMF Landmark Localization

In 2016, Zhang et al. [82] proposed a segmentation-guided partially-joint regression forest (S-PRF) model to automatically localize 15 CMF anatomical landmarks. Later in 2017, Zhang et al. [4] employed a cascaded U-Net architecture to localize 15 CMF anatomical landmarks.

Despite the successes, machine-learning algorithms suffer from two major drawbacks. First, depending only on the local appearance is misguiding, especially when high-morphological variations of the patients are considered. Next, in all the proposed methods, spatial coherence is considered to a degree in the Euclidean space. However it is not possible to learn the spatial consistency of a manifold in the Euclidean space of the medical scan. Hence, in this thesis for the quantification of the medical data, we proposed two different approaches. First, in contrast to the Euclidean space, we proposed a joint segmentation and landmark localization framework in the Geodseic space. Next, we constrained the problem; and without employing the segmentation information we introduced a relational reasoning landmark localization framework.

CHAPTER 3: VOLUMETRIC DATA VISUALIZATION

3.1 Direct Volume Rendering

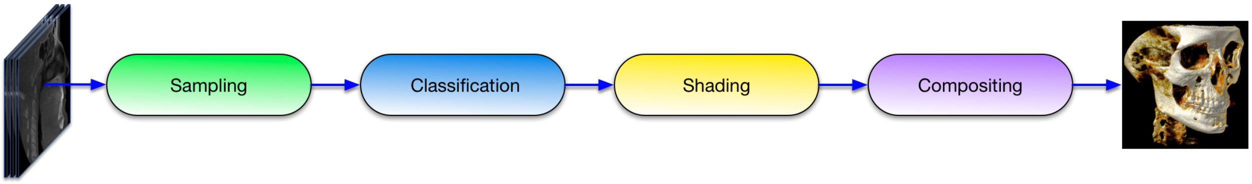


Figure 3.1: Rendering Pipeline.

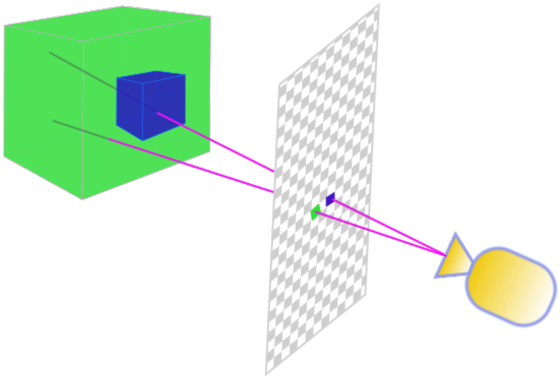


Figure 3.2: Ray Casting.

Direct volume rendering methods refers to the generation of 2D images of a 3D volumetric data set without explicitly extracting geometric surfaces from the data [39]. In the basic algorithm, rays are cast from the camera point to each image pixel. Each ray traverses through the volume data, and accumulates color and opacity values (Figure 3.2) according to an optical model. The core direct volume rendering pipeline is composed of four steps as displayed in Figure 3.1. Volume data is a discrete 3D grid. *Sampling* refers to accessing this discrete grid as if it is a continuous $3D$

data, and computing the value of arbitrary voxels, volumetric element, at any point along the ray by employing tri-linear interpolation. Each sampled voxel is assigned a color and an opacity in the *classification* step. The intensity of the voxel is mapped to a color and an opacity using a *Transfer Function*. Transfer functions can be a simple lookup table or a multi-dimensional function. In the *shading* step, the lighting effects are computed using a lighting model. The lighting computation requires the surface normal which is defined by the gradient. However, computing the differential value at each sample is computationally inefficient. In practice, gradients are computed in a pre-processing step, and used through out the volume rendering process. In the last *compositing* step, the samples along the ray are blended (Figure 3.2). The order of composition depends on the algorithm being applied: front-to-back or back-to-front. In practice front-to-back integration is preferred due to *early-ray-termination* optimization. Once the blended opacity reaches a maximum opacity value (such as 1) the volume at that step is considered to be opaque, it is not possible to see the voxels behind, hence the ray-casting is stopped.

Direct volume rendering assumes that the volumetric data is a semi-transparent emitting medium and each voxel is a tiny entity (like water droplet in the cloud [11]) that interacts with the light passing through it. This interaction can be modeled based on the laws of physics including absorption and scattering.

3.1.1 Optical Models

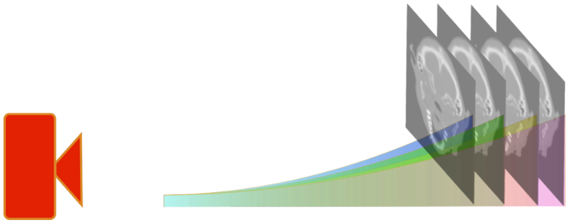


Figure 3.3: Radiant Energy Emission and Absorption

Emission-absorption optical model (Figure 3.6) is sufficient for an insight real-time visualization in the clinical settings. Although, scattering adds a high sense of realism to the rendered image, it suffers from high computational complexity, which is difficult to employ in a real-time application.

A ray cast to the volume is parametrized by its distance t to the camera (camera is assumed to be at the origin 0). Let v_t be the volume point at distance t to the camera, emission of v_t refers to the color of the voxel (c_t), whereas absorption is specified by the extinction (κ_t). Some of the radiant energy emitted by v_t is absorbed by each voxel along the light travel path from v_t to the camera (Figure 3.6). The amount of radiant energy reaching the camera from v_t is decreased by the integral of the absorptions along the path:

$$\hat{c_t} = c_t . e^{-\int_0^t \kappa(\hat{t}) d\hat{t}} \tag{3.1}$$

The cumulative amount of radiant energy received at the camera from the volume along the ray is the integral of the radiant energy along the path:

$$C = \int c_t . e^{-\int_0^t \kappa(\hat{t}) d\hat{t}} dt \tag{3.2}$$

The integral computations are approximated by Reimann sum. Equation 3.1 becomes:

$$\hat{c_t} = c_t.e^{-\sum_{i=0}^{\lfloor t/\Delta t \rfloor} \kappa(i.\Delta t)\Delta t} \tag{3.3}$$

$$e^{-\sum_{i=0}^{\lfloor t/\Delta t \rfloor} \kappa(i.\Delta t)} = \prod_{i=0}^{\lfloor t/\Delta t \rfloor} e^{-\kappa(i.\Delta t)\Delta t} \tag{3.4}$$

$$\hat{c_t} = c_t. \prod_{i=0}^{\lfloor t/\Delta t \rfloor} e^{-\kappa(i.\Delta t)\Delta t} \tag{3.5}$$

The color emitted at the i^{th} sample point in the volume along the ray, C_i is the tri-linear interpolated color of the 26-neighboring voxels around that point. The equation 3.5 becomes:

$$\hat{C_i} = C_i. \prod_{i=0}^{\lfloor t/\Delta t \rfloor} e^{-\kappa(i.\Delta t)\Delta t} \tag{3.6}$$

Opacity at the sample point i, is represented by:

$$A_i = 1 - e^{-\kappa(i.\Delta t)\Delta t} \tag{3.7}$$

Hence the equation 3.4 is simplified to:

$$\prod_{i=0}^{\lfloor t/\Delta t \rfloor} e^{-\kappa(i.\Delta t)\Delta t} = \prod_{i=0}^{\lfloor t/\Delta t \rfloor} (1 - A_i) \tag{3.8}$$

Cumulative amount of radiant energy received along a light travel direction in the discrete volume becomes:

$$\hat{C} = \sum_{i=0}^{n} C_i . \prod_{j=0}^{\lfloor t/\Delta t \rfloor} (1 - A_j) \tag{3.9}$$

Equation 3.9 can be implemented using back-to-front or front-to-back algorithms. In the back-to-front implementation, the integration starts at the farthest voxel to the camera in the ray direction, and iterates towards to the camera. Whereas, in the front-to-back, integration starts at the closest voxel to the camera in the ray direction, and iterates in the reverse direction of the camera. However, it is not possible to employ *early ray termination* optimization in the back-to-front implementation.

Front-to-back (See Algorithm 1) ray-casting has been implemented independently using WebGL, OpenGL and OpenCL languages. WebGL has been used for the standalone web-based visualization [57]. Both OpenGL and OpenCL have been used independently for the implementation of the client-server model [54]. OpenGL has been used for the implementation when using the local server, whereas OpenCL has been used when using the remote server.

The visual quality obtained by both implementations are similar, however OpenCL rendering has been found to be a little slow compared to the OpenGL version. The difficulty in getting OpenGL device context for remove servers, has been the main factor for using OpenCL implementation for the remote servers.

3.1.2 Lighting

Lighting refers to modeling the interaction of each voxel with the light rays. Once the light rays are emitted from the light source, they get reflected and scattered around in the scene until some

Algorithm 1 Front-to-back Rendering.

```
procedure FRONT_TO_BACK($camera, rayStart, rayEnd, uPhysicalDim$)
    $currentPos \leftarrow rayStart$
    $rayDirection \leftarrow normalize(rayEnd - currentPos)$
    $stepSize = uPhysicalDim/MAX\_STEPS$
    $stepSzScaledRayDirection \leftarrow rayDirection \times stepSize$
    $resultColor \leftarrow 0$
    for $i$ less than $MAX\_STEPS$ do
        $ijk \leftarrow physical\_space\_to\_image\_space(currentPos)$
        $gradient \leftarrow sample(uGradientSampler, ijk).rgb$
        $intensity \leftarrow sample(uVolumeData, ijk).r$
        $color.rgba \leftarrow sample(uTransferFunction, intensity)$
        $color \leftarrow blinnPhong(camera, currentPos, color, gradient)$
        $resultColor \leftarrow resultColor + ((1 - resultColor.a) \times color)$
        if $resultColor.a > 0.95$ then                ▷ early ray termination optimization
            $break$
        $currentPos += stepSzScaledRayDirection$
    return $resultColor$                ▷ accumulated color and opacity is returned
```

of them reach the camera. However, implementations of the full scattering and the inter-reflection algorithms are computationally expensive. In the medical volume rendering implementations, full scattering and inter-reflections are not employed. Often a simple lighting model such as Phong model [41] is sufficient. This model computes the lighting using three computationally simple components: *ambient*, *diffuse* and *specular* lighting. Ambient lighting approximates inter-reflection and scattering by a user defined constant parameter. Diffuse lighting models the matte material behavior, whereas specular lighting models the shiny material behavior. Specular lighting adds visual realism to the rendered volume (See Figure 3.4). In the specular reflection, in contrast to the equal reflection to all directions of the diffuse reflection model, the light is reflected more in some particular directions. Unlike the diffuse lighting where the light is uniform in all directions, in the specular lighting, the light intensity varies with the direction. The variation is modeled by the Phong [41] or the Blinn-Phong [40] equation. Implementation for this thesis employs the Blinn-Phong illumination model.

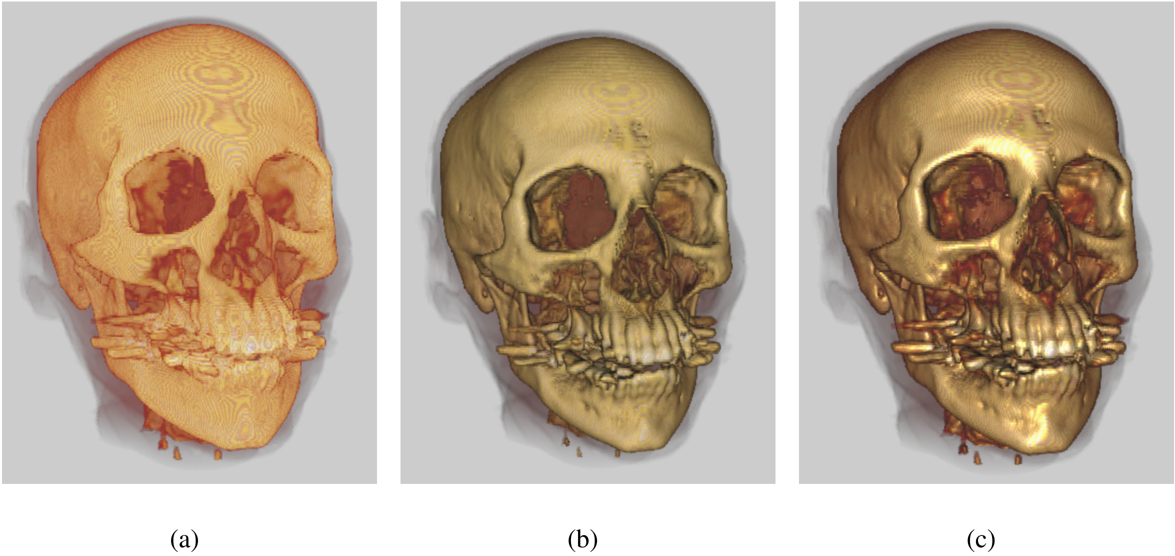
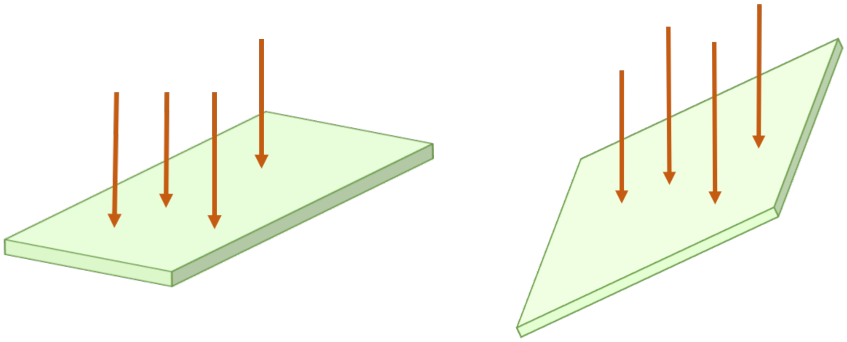


Figure 3.4: (a) Ambient Lighting (b) Ambient + Diffuse Lighting (c) Ambient + Diffuse + Specular Lighting.

In computer graphics applications, the lighting models that we discussed above are used for computing the reflection from the opaque surfaces. Each surface is assigned reflection properties using the reflectivity parameters; K_a, K_d and K_s for modeling the ambient, the diffuse and the specular reflections respectively. The specular highlights are controlled by the fourth parameter: shininess (a).

Ambient color of the surface is computed according to the ambient reflectivity and ambient intensity:

$$C_a = K_a \times I_a \tag{3.10}$$



(a) (b)

Figure 3.5: Light source and surface angles a) Perpendicular b) Other.

For the diffuse surfaces, the light received per unit area of the surface is assumed to follow the Lambert's law, which is the angle between the surface and the light direction (Figure 3.5). When the surface is perpendicular to the light source, it receives more light per unit area compared to the surfaces at the other angles. The cosine of the angle of the surface and the light direction is computed by employing the dot product of the surface normal and the light direction. Hence the diffuse component of the reflected light is the product of the diffuse reflectivity, the diffuse intensity and the cosine of the angle between the surface normal and the light direction:

$$C_d = K_d \times I_d \times (N \cdot L) \tag{3.11}$$

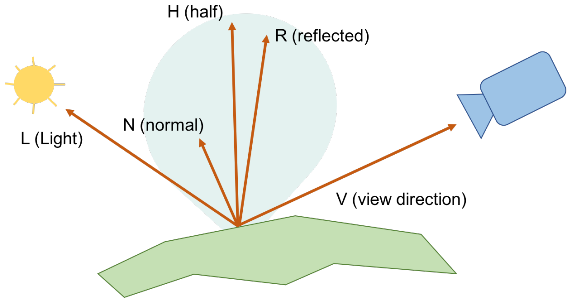


Figure 3.6: Blinn-Phong Model.

In contrast to the matte appearance of the diffuse lighting, the specular lighting adds a shiny appearance to the surfaces (Figure 3.4). In the Phong equation, the specular component of the reflected light depends on the angle between the reflected light and the view direction. When the angle is zero, that means the view direction matches with the mirror reflection direction, the reflection strength is maximum, and the strength of the reflection degrades as the angle increases. The Phong equation is as follows:

$$C_s = K_s \times I_s \times (V \cdot R)^a \tag{3.12}$$

The Blinn-Phong equation, a small variation of the Phong equation, is proposed as a physically correct model of the specular reflection. This model simply replaces the reflection vector of the Phong equation with the half vector. The half vector, H, is the average of the light and the view directions. Hence, the Blinn-Phong equation is:

$$C_s = K_s \times I_s \times (N \cdot H)^a \tag{3.13}$$

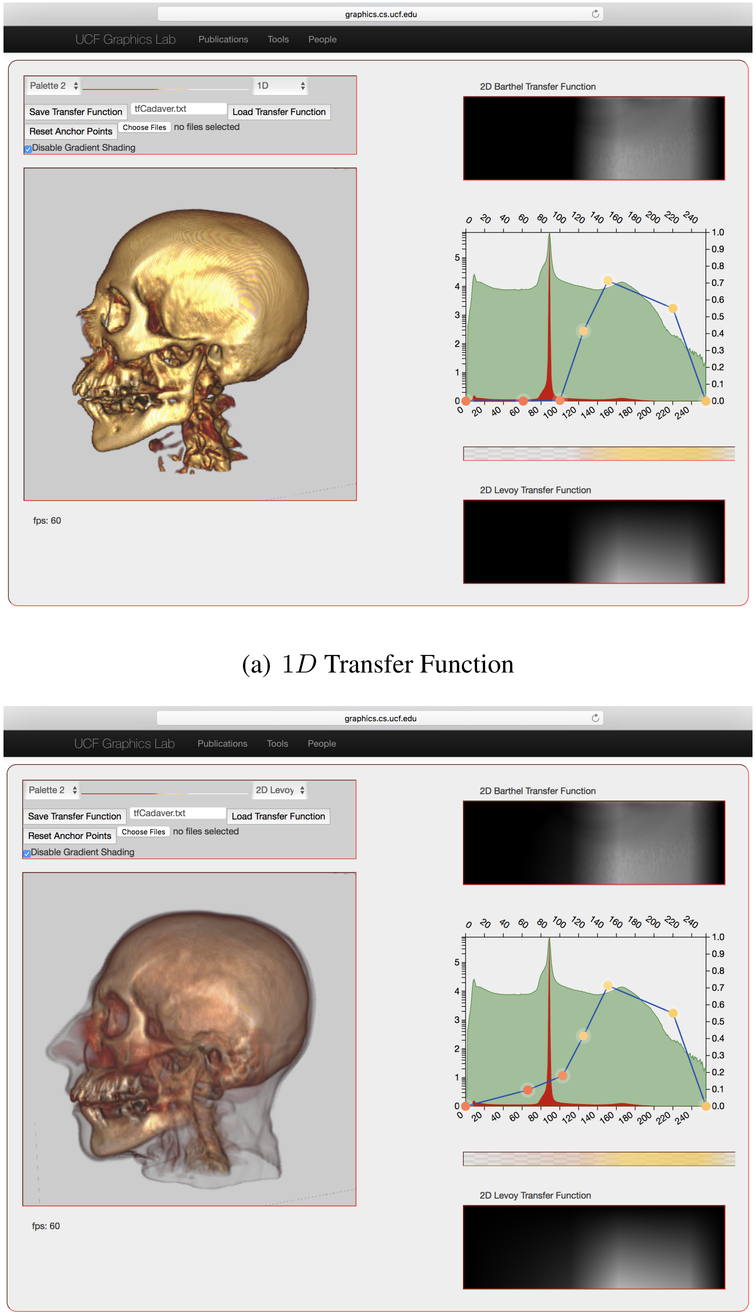
In this thesis, it is assumed that there is a light source at the camera location, hence the light vector is the vector from the center of the volume to the camera location.

3.2 Web-based Medical Data Visualization

WebGL (Web Graphics Library) is a cross-platform, royalty-free web standard for 3D graphics API. It is based on OpenGL ES, and it runs on any compatible web browser without using any plug-in. WebGL enables the GPU-accelerated graphics applications on the web page canvases. WebGL programs are composed of two main parts: the JavaScript code and the shader code (OpenGL Shading Language (GLSL)). Khronos Group maintains the WebGL [83]. At the end of 2007, both Mozilla and Opera developed their WebGL implementations. Today, all major browser vendors Apple (Safari), Google (Chrome), Microsoft (Edge), and Mozilla (Firefox) support WebGL.

VOLREN [57] is a web-based medical data visualization application developed using WebGL (`http://graphics.cs.ucf.edu/tools/VOLREN/`) (See Figure 3.7). We employed front-to-back ray casting approach, hence applied early-ray-termination optimization. For lighting, we applied the Blinn-Phong [41] lighting model. The X Toolkit [84] is used to parse the medical files. The user interface allows selecting among $1D$ or $2D$ transfer functions. We employed $D3$ [85] to develop the customizable $1D$ transfer function. Zoom, rotation and translation functionalities are implemented for both the mouse and the touch events which makes the application accessible at the non-touch displays by the mouse events, and at the touch displays by the touch events.

VOLREN enables interactive and customizable rendering of the medical data on any portable device with WebGL supported browser. However, WebGL applications are limited with the GPU power of the device the software runs on. When the rapid increase in the medical data sizes, and the limited GPU capabilities of the low-end portable devices are considered, it is not possible to achieve interactive rates by using standalone WebGL based software.


graphics.cs.ucf.edu

Palette 2

1D

tfCadaver.txt

no files selected

☑ Disable Gradient Shading

fps: 60

(a) $1D$ Transfer Function

graphics.cs.ucf.edu

Palette 2

2D Levoy

tfCadaver.txt

no files selected

☑ Disable Gradient Shading

fps: 60

(b) $2D$ Transfer Function [39]

Figure 3.7: VOLREN [2] Transfer Functions

3.3 Out-of-core Medical Scan Visualization

Many applications in a wide range of domains such as medicine, biology, physics, and engineering benefit from the visualization of the volumetric data at the interactive rates. The performance of the GPUs has been increasing rapidly in the recent years and their parallel processing power can now be harnessed in the visualization applications using OpenCL/Cuda or OpenGL/DirectX APIs [86]. However, volume rendering is still challenging for larger volumes that exceed the resources available on the rendering hardware. This is particularly true for the mobile clients. Although there are some large-scale volumetric visualization tools available such as Vaa3D-TeraFly [87], none of them fulfill the requirements especially for the mobile clients. Hence comes the need for a large volume interactive visualization architecture that can scale in the performance by using the server-based computing, yet deliver the volumes to the low-end (e.g. mobile) devices.

To be interactive requires rendering the volume at the interactive rates (generally more than five frames per second). The computation cost associated with a volume renderer, makes it nearly impossible to guarantee the interactive frame rates independent of the computing power of the user's device. To address this problem, we have developed a client-server framework; where the server is responsible for the compute/storage/memory intensive tasks, such as volume processing and rendering the view. The client is responsible for the interaction, the transfer function manipulation and the display of the image frames rendered by the server (see Figure 3.8). The appropriate choice of the server hardware configuration and a decent network connectivity will deliver the interactive frame-rates. Any browser-based device, serving as a client, will allow the user to interact with the volume and interactively view the image rendered by the server. To our knowledge, ours is the first web-based out-of-core volume visualization system.

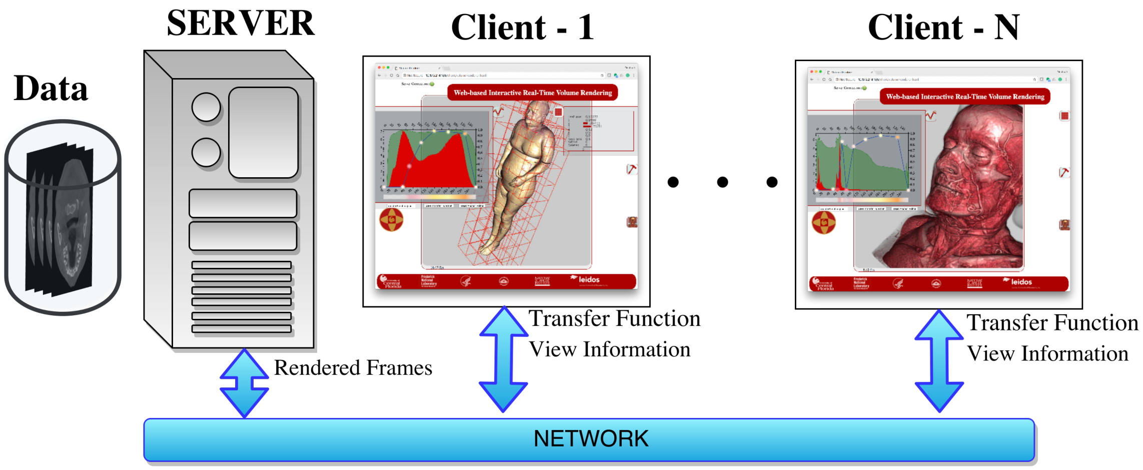


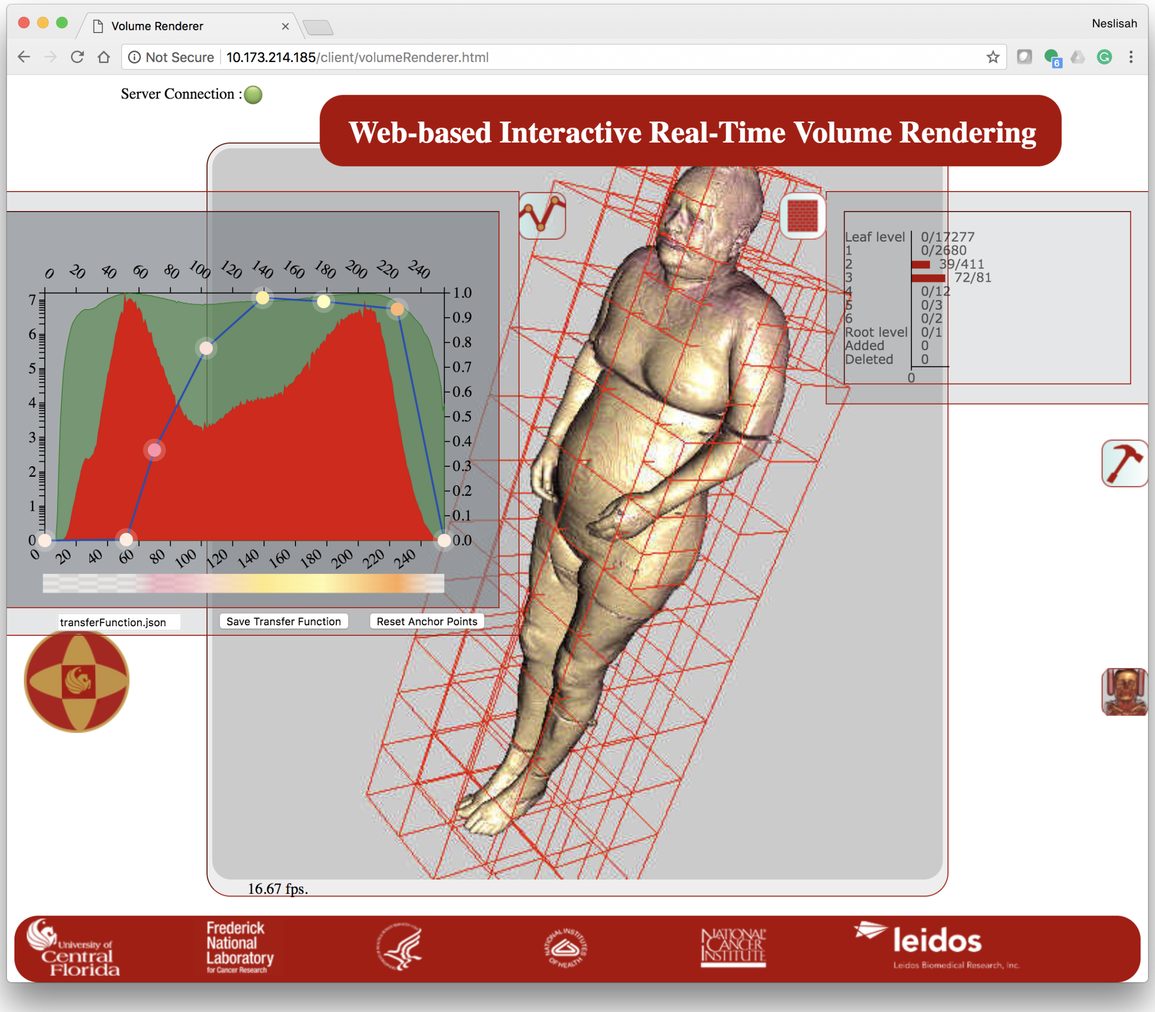
Figure 3.8: Client-Server architecture.

On the server side, we employed front-to-back ray casting approach, hence applied early-ray-termination optimization. For lighting, we applied the Blinn-Phong [41] lighting model. The Insight Segmentation and Registration Toolkit (ITK) [88] is used to parse the medical files. Both OpenGL and OpenCL have been used independently for volume rendering. OpenGL has been used for the implementation when using the local server, whereas OpenCL has been used when using the remote server. The difficulty in getting the OpenGL device context for the remote servers, has been the main factor for using the OpenCL implementation for the remote servers.

On the client side, we used HTML5 and Javascript to develop the user-interface. The user interface allows customizing the transfer function using $D3$ [85]. Zoom, rotation and translation functionalities are implemented for both mouse and touch events which makes the application accessible at the non-touch displays by the mouse events, and at the touch displays by the touch events.

3.3.1 Server

On the server side, the volume dataset is partitioned into an octree hierarchy of bricks, and as needed, before rendering each frame, a view dependent candidate set of the bricks from this hierarchy is computed and transferred to the GPU along with an index texture describing the mapping of the available bricks to the GPU memory. The volume is first partitioned into bricks. The brick size is set to be some power of 2-cube (say a cube of size $32 \times 32 \times 32$). Processing of the volume data, which includes partitioning of the volume data into the octree hierarchy and pre-computing gradients, is a time-consuming task and hence is, carried out in the GPU in a preprocessing step. The results are stored for later use. In the pre-processing stage, in addition to the volume data and the octree structure, histograms and lookup tables are also computed and stored. Since an octree inherently requires the size to be a power of 2 cube, processing any rectangular, non-power of 2 volume, requires us to create non-existing (hereto called "ghost") bricks, that are not physically stored anywhere, but are required for the octree handling. We use a lookup table to keep track of such ghost bricks. Depending on the data size, the preprocessing step can be very time consuming. For example: preprocessing Visible Human Project female dataset takes around 1 hour to preprocess data on Intel Core i7-4770K Processor and GeForce GTX 780 GPU, with 3GB GPU and 15.6GB CPU Memory. In our system, every brick in the octree hierarchy is assigned a unique id consisting of a two-component tuple (level#, index), where index is the flattened 3D index of the brick in the volume. Depending on the system memory capacity, bricks are stored in physical or virtual CPU memory. For each rendered frame (triggered by interactive viewing and/or transfer function update) we traverse through the octree hierarchy in top-down order to compute a candidate brick set (see Figure 3.9). The resolution of the bricks in the candidate set are chosen such that the number of voxels on the face of the brick closely matches with the number of the pixels in the foot-print of the view-dependent projection of the brick bounding volume on the display window.



Server Connection : ●

Leaf level	0/17277
1	0/2680
2	39/411
3	72/81
4	0/12
5	0/3
6	0/2
Root level	0/1
Added	0
Deleted	0

16.67 fps.

Figure 3.9: View dependent brick computation.

The required bricks′ resolutions are identified according to their locations by traversing the octree in the breadth-first order. The allocated GPU memory is organized as a cubic volume whose size is an exact multiple of brick size. Thus each brick in the candidate set can be assigned a unique location in the GPU resident volume.

Once the candidate brick set computation is completed, an index table is created to map the voxels encountered at render time to the appropriate brick in the GPU cubic volume. The size of the index table is set to be equal to the number of the bricks at the leaf level. The candidate set is processed one brick at a time in the order in which they appear, and a tuple, composed of the octree level of the candidate brick and its location at the GPU memory, is assigned to the cells of the table that corresponds to all the descendant leaf bricks of the candidate brick. The index table is stored as a two-channel 2D integer texture. Once the index table construction is completed, view dependent candidate bricks and the index texture are sent to the GPU, and the front-to-back ray-casting algorithm is executed. During the ray traversal in the front-to-back ray casting, for each point along the ray in the volume, its corresponding CPU brick id is computed, and the index texture pixel for this brick id is read to find the mapped GPU brick id and its octree resolution. If the voxel maps to a brick, then the offset for the voxel data in the corresponding GPU volume memory is computed, and the voxels properties (emission, its opacity and gradient) are read. If the voxel does not map to any brick, computation continues to the next voxel until either the ray exits the bounding volume or early ray termination occurs. Early ray termination is applied when opacity of the voxel reaches 95 percent or above.

All the server side implementations are carried out in C++ using the Boost library. To speed-up the brick lookup, the octree of image bricks is stored in a memory-mapped file. The GPU programming for octree construction and gradient computation is carried out using OpenCL. If the server is running locally then OpenGL implementation of volume ray casting is used otherwise OpenCL implementation is used. For the former (i.e. locally run server) the user has an option to run in OpenCL also. However, our OpenGL implementation runs faster. Front-to-back ray casting is implemented using a two-pass rendering, where the first pass computes the ray exit points in the volume, and the second pass executes the actual front-to-back ray casting and rendering. The voxel gradients of the volume are used as the normal for shade computation using a Phong lighting model.

3.3.2 Client

We implement a web-based client to make our system accessible from a wide variety of platforms without any setup requirement. The client and server connect over WebSockets[89]. Websockets are chosen due to their full-duplex capabilities on a single socket connection. Although in our implementation, clients trigger most of the actions in the server, full-duplex communication is required as for instance when preprocessing is completed, server needs to send a message to the appropriate client. Our client handles display and interaction with the volume and interactive transfer function creation. Requests initiated by the client include load new file request, transfer function update, and camera update, among others. The server renders the volume and sends back the rendered frame to the client through the WebSocket connection [89].

WebSocket server side is developed using Boost Library. The server runs $n + 1$ threads running in parallel, where n is the number of simultaneous client interfaces connected to the server. The first thread is used for listening for socket requests and creating a new thread for each connection request. Each successive thread created, handles the rendering requests of the client on the other end of the connection, and transmits the rendered frame to the client.

Client requests are encoded in the JSON format. Each render request has an action tag; so the server knows what to perform when it parses the request. There is a queue shared between the listening thread and each client thread. When the listening thread receives the request in JSON format, it parses the request, and puts the appropriate action request in the appropriate shared queue. Client threads, on the other hand, polls the shared queue, and when there is an action request pushed to the queue, the client thread performs the rendering request and transmits back the rendered frame to the appropriate client through the open connection. The rendered frames are transmitted to the clients as binary images.

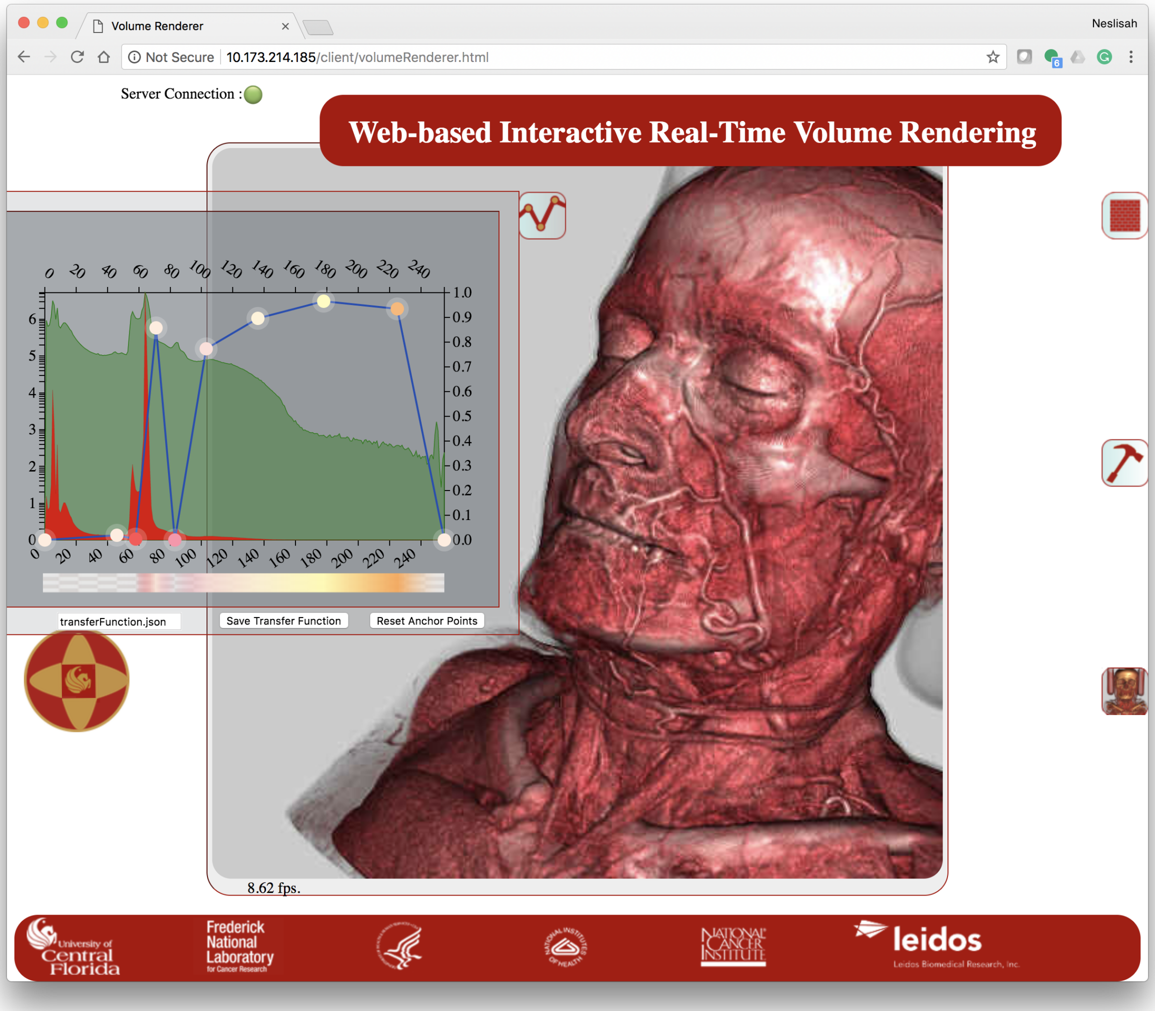

Volume Renderer
Neslisah
Not Secure
10.173.214.185/client/volumeRenderer.html
Server Connection :
Web-based Interactive Real-Time Volume Rendering
0 20 40 60 80 100 120 140 160 180 200 220 240
0 1 2 3 4 5 6
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
transferFunction.json
Save Transfer Function
Reset Anchor Points
8.62 fps.
University of Central Florida
Frederick National Laboratory for Cancer Research
National Institutes of Health
National Cancer Institute
leidos
Leidos Biomedical Research, Inc.

Figure 3.10: Customizable transfer function.

D3 is used for implementing an interactive Transfer Function editor on the web page (Figure 3.10). The Transfer Function is composed of an array of anchor points, where each anchor point has x, y, r, g, and b float values. x represents gray scale voxel value, y represents transparency of the anchor point, and r, g, and b refer to red, green and blue color components of the mapped color value. The user can interactively add/delete/modify the opacity and the color of an anchor point. The user is allowed to save and retrieve the transfer function.

When client makes a load a new volume request, if the volume is not already preprocessed, then the preprocessing is automatically triggered. When preprocessing is completed, server sends message to the requesting client about completion of preprocessing stage. The bricks are stored sequentially in a file. At the time of rendering, when a brick at a certain level is required, its data is fetched from the appropriate position. Since spatially closer bricks are more likely to have similar projected area and belong to the same octree level, they appear adjacent to each other, and hence fetching them using the Memory Mapped File (available in the Boost library) is fast. We obtain an acceptable visual quality, which is highly dependent on the transfer function, at an average frame rate of 10Hz.

The profiling results reveal that GPU-CPU communication and network connection capacity are the two major bottlenecks of the application. Hence, management of bricks on the GPU is a crucial part of our algorithm. Efficient mapping of the available brick slots in the GPU for the adaptively-computed, view-dependent brick candidates is critical to the efficiency of the out-of-core algorithm. In addition, considering the frame-to-frame coherence, and transferring just the missing bricks to the GPU also resolves this problem to a degree. Although, the implementation of the frame-to-frame coherence for two very big sets of data is not a computationally fast algorithm, the loss is much lower than the gain [54].

To the best of our knowledge, our client-server application is the first out-of-core volumetric medical data visualization software which enables low-end devices with limited computing capabilities such as smart-phones or mobile tablets to visualize large volumetric data at interactive rates. Local server-based systems are designed to serve some limited number of clients. However, the GPU-cloud servers provide highly scalable solutions.

In the clinical point of view, gathering the $3D$ visual clues at interactive rates is very crucial. However, in addition to the visual clues the clinicians need to make quantitative measurements to diagnose, to determine the severity of diseases and to prepare the treament/surgery planning. Hence, visualization, the indispensable part of a medical image analysis framework, is not by itself sufficient in the clinical point of view. Therefore, additional smart quantification algorithms are needed to provide a complete, and clinically useful image analysis framework.

CHAPTER 4: MANDIBLE SEGMENTATION

This chapter is based on my paper entitled “Robust and Fully Automated Segmentation of Mandible from CT Scans” published at the proceedings of the IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017). Copyright ©2017 IEEE.

4.1 Introduction

In the United States, there are more than 17 million patients with congenital and developmental deformities in the anatomical area of the mouth, jaws, face, and skull, which is called the cranio-maxillofacial (CMF) region [8]. Post-traumatic defects, deformities of the temporamandibular joints, defects of tumor ablation, and congenital diseases are some of the leading causes of these deformities [8]. The number of patients who need orthodontic treatment is far beyond this number. Among the CMF conditions, the mandibular region (Figure 5.1(a)) is the part most often deformed or injured. For instance, 76% of the skeleton traumas affects the mandibular region [9].

Hence, we started our quantification research in the mandibular region and developed a novel and fast algorithm to fully-automatically and precisely segment the mandible bone in CT scans.

We consider the mandible segmentation procedure as consisting of two complementary tasks: recognition and delineation. While recognition is the process of determining roughly “where” the object is and distinguishing it from other object-like entities in the image, delineation is the act of defining the spatial extent of the object region in the image [90]. In our previous publication, we have shown the significance of having a successful recognition step for an efficient and accurate segmentation process [90, 91]. In this study, an efficient recognition step avoids potential leakages in the delineation by constraining the segmentation region into a tight search space.

4.2 Step I: Recognition of mandibular bone

We assign a probability score for each slice (in a multi-view setting) if it includes the mandible bone. Then, we combine these views (sagittal, axial, and coronal) to improve probabilistic determination of the mandible location. Labeled slices (0 or 1) are used to train an Random Forest(RF) based regression algorithm. Probability scores from each view are fused, and we obtain a continuous output scale for which probability value of 0.5 and higher for a particular slice is considered to include mandible bone. Instead of multi-view setting, one may directly use 3D implementation of the RF based classification. However, particularly in CMF analysis with low dose CT scans, or CT scans with asymmetric pixel sizes, it is reasonable to use the limited information from certain slices and views.

4.3 Step II: Gradient-based fuzzy connectivity

After identifying the object of interest through a bounding-box, tightly enclosing the mandible, we propose to use the gradient-based Fuzzy Connectivity(FC) algorithm to minimize leakage possibilities while delineating the mandible. FC has shown to be very robust compared to the other algorithms such as Graph-Cut (GC) and Level Set (LS) [92]. However, the FC algorithm requires a pre-defined mean and standard deviation of object's intensity distributions, and since there are bones, joints, and other artifacts in the vicinity of mandible sharing strong intensity similarities, it is still possible for the delineation algorithm to leak. Thus, we re-parametrize the FC algorithm to accept only one parameter that is calculated through a gradient image instead of the original gray-scale CT scan. By this change, we emphasize the boundary locations in delineation.

Let a topology on an image be given in terms of an *adjacency* relation (μ_α) such that if p and q are α-adjacent to each other, then $\mu_\alpha(p, q) = 1$, '0' otherwise. In practice, we set $\alpha = 26$ for 3-D mandible bone analysis. While affinity is intended to be a local relation, a global fuzzy relation, called fuzzy connectedness, is induced on the image domain by the affinity functions as described in detail in [93]. This is done by considering all possible paths between any two voxels p and q in the image domain, and assigning a strength of FC for each path. The level of the FC between any two voxels p and q is considered to be the maximum of the strengths of all paths between p and q. An *affinity relation* κ defines the strength of spatial and intensity-based similarity between two voxels and is the most fundamental measure of local hanging togetherness of the nearby voxels. For a path π, which is a sequence of the voxels $\langle p_1, p_2, ..., p_l \rangle$ with every two successive voxels being adjacent, given a *fuzzy affinity function* $\mu_\kappa(p_i, p_{i+1})$, the strength of the path is defined as the minimum affinity along the path [91, 93]:

$$\mu_{\mathcal{N}}(\pi) = \min_{1 \leq i < l} \mu_\kappa(p_i, p_{i+1}). \tag{4.1}$$

Then, the strength of connectedness $\mu_{\mathcal{K}}(p, q)$ between any two voxels p and q is the strength of the strongest path between them as

$$\mu_{\mathcal{K}}(p, q) = \max_{\pi \in \mathcal{P}(p,q)} \mu_{\mathcal{N}}(\pi), \tag{4.2}$$

where $\mathcal{P}(p, q)$ denotes the set of all paths between p and q. A FC object $\mathcal{O}$ (i.e., mandible) in an image can be defined for a predetermined set of seeds S. Note that in our algorithm, there is only one seed necessary to initiate FC delineation and we solve this issue by selecting the voxel with highest intensity value and largest connected component in the bounding-box. The final object is obtained by thresholding over the fuzzy object $\mathcal{O}$ for the strength of the connectedness.

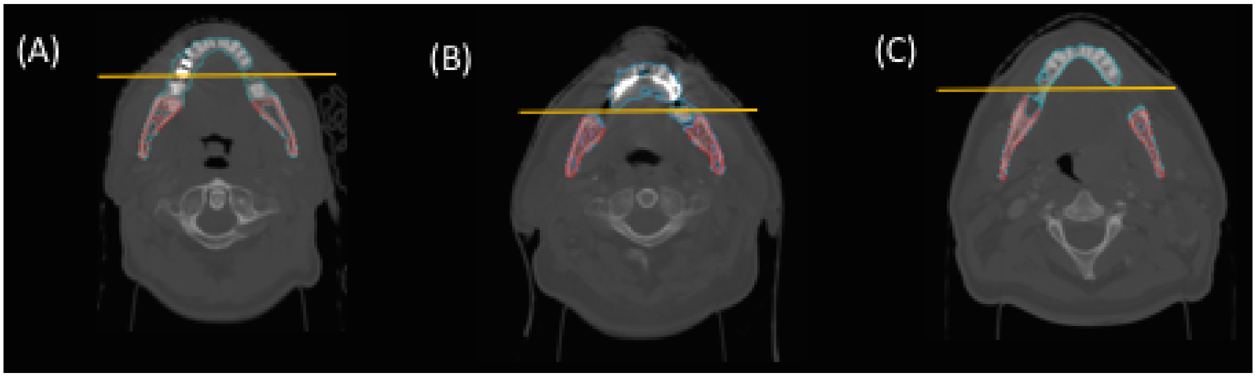


Figure 4.1: Separation of teeth and mandible, ground-truth (red) and FC (blue).

4.4 Step III: Boundary refinement for final segmentation

We present a new boundary refinement algorithm to handle the leaks. Although it is small, leakage can still occur during this separation due to strong overlap between intensity distribution of these structures (Figure 4.2). Based on anatomical knowledge, we design a state-machine, composed of five states: initial state, base state, teeth state, leak state, and ending state (Figure 4.2). The number of connected components and their sizes in consecutive axial slices are used to switch from one state to another. Briefly, we utilize a state-machine heuristic approach to track potential leaks, and then refine those regions by incorporating the domain knowledge: axially, mandible regions include a few connected components with small sizes and this information is quite robust even with highly deformed jaws (the initial state in shown in Figure 4.2-A). Following, the connected components start merging and become large in physical size (base state in Figure 4.2-B). Next, the number of connected components increases as teeth are introduced (teeth state in Figure 4.2-C). In the teeth state, *k-means* clustering with two classes is used to separate teeth from mandible. This rough separation allows us to automatically localize teeth, located in the front portion of the axial slice, and mandible at the back (See Figure 4.1). In the leak state, we use the change in width and height of delineated mandible to track *abrupt changes* to detect leak areas (leak state in Figure 4.2-E and Figure 4.3).

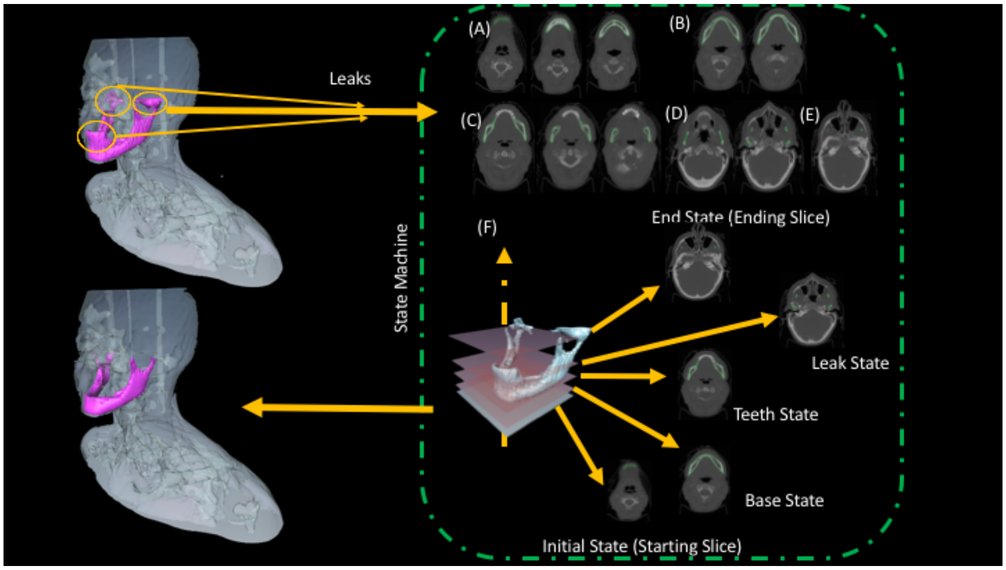


Figure 4.2: Boundary refinement using a five step state-machine.

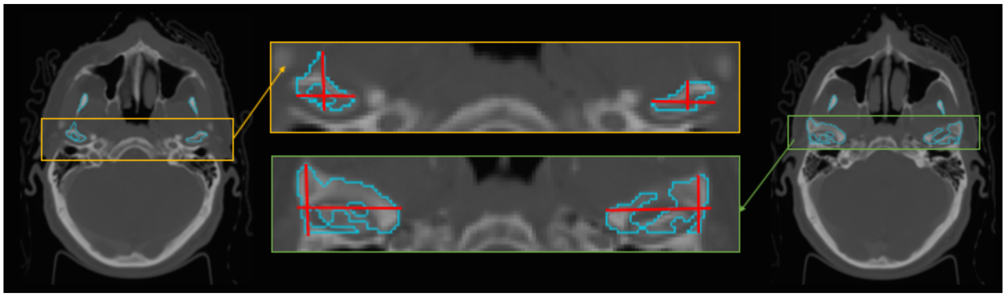


Figure 4.3: Leak state, FC (blue) and the width and height of delineated mandible (red). Abrupt change in width or height in consecutive slices means a leak in the mandible skull joint.

4.5 Experiments and Results

Data: We clustered MICCAI Head-Neck Challenge 2015 data set [94] into 3 groups, based on the severity of artifacts on the CT data as: No or Low, Medium, and High. This was done to experimentally show robustness of our proposed system with respect to the varying amount of artifact. While 17 volumes were classified as no or low artifact, 4 of them were labeled as medium, and 19 of them were classified as high amount of artifacts. The in-plane resolution of the CT scans was 1.12×1.12mm, and the slice thickness was 3 mm.

We conducted our experiments on a low-end regular laptop, MacOS-X ($2.6GHz$ Intel Core-i5) with $8GB$ CPU memory. Three CT volumes were randomly selected for training and the rest were used for testing in RF based recognition experiments. Based on a commonly used detection/recognition evaluation metric, called the intersection of union (IoU), we have performed a detection accuracy of more than 96%. For delineation accuracy, we have used dice similarity coefficient (DSC) as well as a modified version of Hausdorff Distance (HD) to quantify boundary shape mismatch. The delineation performances are summarized for all CT scans and with respect to the varying amount of CT artifacts (low, medium, and high) in Figure 4.4.

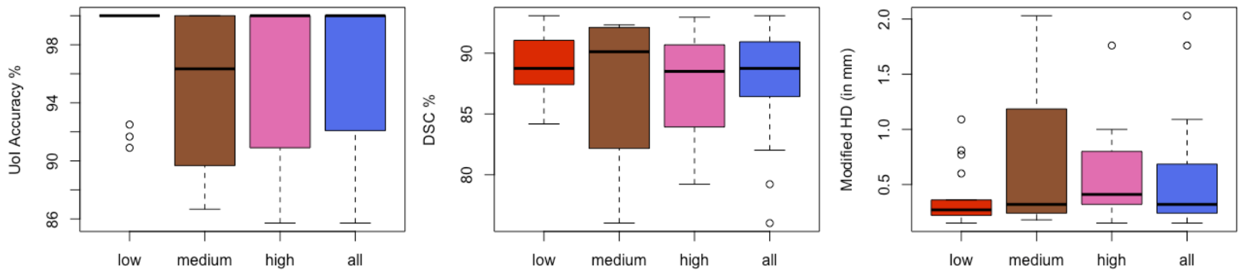


Figure 4.4: IoU, DSC, and modified Hausdorff Distance (HD) metrics are shown for evaluation of detection and segmentation methods with respect to varying amount of artifact (x-axis: low, medium, high, and overall). Note that, high IoU indicates better recognition, lower HD and higher DSC scores indicate accurate segmentation.

Compared to the existing work of [95], the winning algorithm of the MICCAI 2015 Head-Neck segmentation challenge, our algorithm provided similar DSC and HD metrics for all CT scans regardless of the amount of artifacts, confirming the robustness of the proposed method. Furthermore, we claim at least two superiorities: **(1)** our algorithm is completely data-driven, **(2)** regardless of the amount of artifacts, we obtained similar accuracies for mandible segmentation, hence the proposed algorithm is robust.

4.6 Discussions and Concluding Remarks

In this work, we develop a data-driven, robust, and accurate method for automatic mandible detection and delineation from CT scans. The proposed approach is based on RF regression algorithm for determining rough location of the mandible, and region-based segmentation algorithm with automatic initialization. We integrate anatomy knowledge in the final stage as a state-machine procedure. With this refinement, fine-tuning of the delineation is performed successfully regardless of the amount of CT artifacts and high anatomical variability in the mandible bones. This verifies the robustness of the overall system. By using publicly available segmentation challenge data (MICCAI 2015- Head-Neck Data Set), we tested and evaluated the performance of the recognition and delineation steps of the proposed framework, and obtained similar accuracies compared to the work of [95] but with superiority in robustness, efficiency, and fully-automatic nature.

Even though the proposed algorithm is promising in several ways, the following limitations of our work should be noted as well: performance of our algorithm is not tested in CT scans with bone fractures or bone-loss in or nearby the mandible. Although these cases may be rare in this particular settings, it may be desirable for clinicians to handle such cases (challenge data-set is pertaining head-neck cancer) with the software tool as well. Furthermore, in our study, we integrated expert knowledge on human anatomy (specifically mandible) in state-machine model for the refinement step. However, for congenital diseases and trauma cases, modeling expert knowledge may not be accurate for leak tracking. Therefore, we continued our research using deep learning based algorithms.

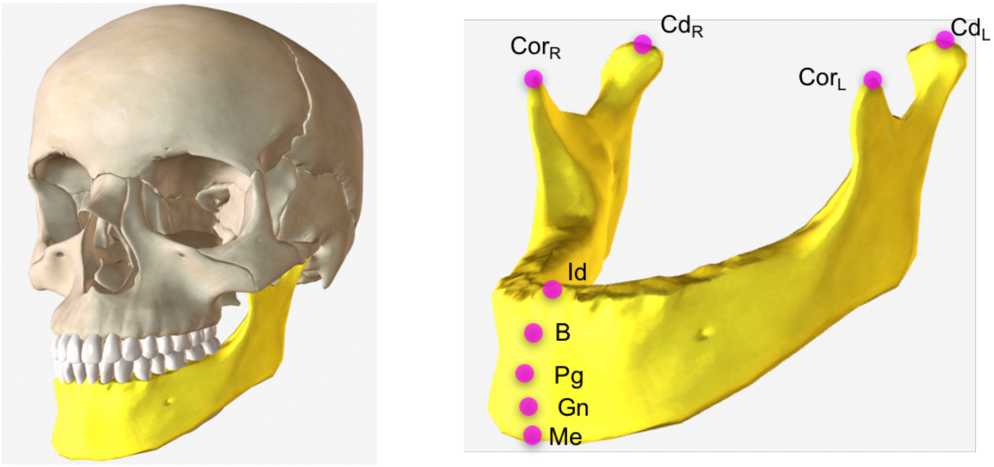
CHAPTER 5: DEEP GEODESIC LEARNING FOR SEGMANTATION AND ANATOMICAL LANDMARKING

This chapter is based on my paper entitled “Deep Geodesic Learning for Segmentation and Anatomical Landmarking” published at the IEEE Transactions on Medical Imaging, 2018. Copyright ©2018 IEEE.

5.1 Introduction

The mandible is the lower-jaw bone, and it is the only mobile bone in the CMF region. It is the largest, the strongest, and the most complex bone in the CMF region that houses the lower teeth and canals with blood vessels and nerves. Due to its complex structure and the significant structural variations of the patients with CMF disorders, segmentation and landmark localization in the mandibular region is a very challenging problem (See Figure 1.1). Although, there are efforts with very promising performances, speed and accuracies [82, 4, 96, 97], the literature still lacks a fully-automated, fast and generalized software solution in response to wide range of the ages of the patients, deformities, and the imaging technology artifacts. Hence, the current convention used in the clinics is either manual segmentation and annotations, or semi-automated with computer software supports (in alphabetical order) 3dMDvultus (3dMD, Atlanta, Ga), Dolphin Imaging (Dolphin Imaging, Chatsworth, Ca), and InVivoDental (Anatomage, San Jose, Ca).

The segmentation and landmark localization procedure is not trivial due to large morphological variations among different patients, disease burden, and inevitable image artifacts including dental fillings, orthodontic wires, bands, and braces. Towards a solution to this challenging problem, there have been many attempts in the literature; however, most of the studies that have addressed



(a) Mandible

(b) Anatomical landmarks on the mandible

Figure 5.1: a) Skull and Mandible (colored in yellow), (b) Anatomical landmarks on the mandible: Menton (Me), Gnathion (Gn), Pogonion (Pg), B Point (B), Infradentale (Id), Condylar Left (Cd_L), Condylar Right (Cd_R), Coronoid Left (Cor_L), and Coronoid Right (Cor_R)

.

the mandible segmentation utilize statistical shape models [98]. Statistical shape models are specifically insufficient especially when large morphological variations such as bone loss or bone fractures are considered. In our previous study (Chapter 4) we also employed expert knowledge in the refinement step which limits the performance of the software in such abnormal cases.

In this stage of our research we employed a data set which includes patients with congenital diseases with extreme variations in CMF bones, and patient population is highly diverse: a wide range of ages (9 to 50 years old), imposing high anatomical variations apart from the diseases. The following image based variations have been confirmed in our database, too: aliasing artifacts due to braces, metallic surgery nails, dental fillings, and missing bones or teeth. The overarching goal of our study is to develop a fully-automated image analysis software for mandible segmentation and anatomical landmarking on the mandible. Our proposed solution is based on a newly designed deep learning architecture and trained in an end-to-end manner. To help clinicians in the measurement process, we include the landmarking process as a part of the segmentation algorithm to make certain geometric measurements accurately, easier, and faster. Our proposed novel deep learning algorithm includes three inter-connected steps (See Figure 5.2 for overview of the proposed method). In the first step, we propose a convolutional neural network (CNN) based segmentation engine for mandibular bone extraction from 3D CBCT scans. In Step 2, we present a learning based geodesic map generation algorithm for each anatomically defined landmark on the mandibular. Finally, in Step 3, inspired by recurrent neural networks (RNN), we demonstrate an LSTM (long short term memory) based algorithm to learn the relationship among anatomical landmarks distributed on the same bone. We utilize both segmentation results and geodesic maps of each landmark in this step. The proposed system is trained in a fully automated fashion, and run over the Geodesic space of the landmarks instead of Euclidean space, which makes it attractive for manifold learning applications too.

The overview of the proposed approach is shown in Figure 5.2. Overall, we solve the segmentation and landmarking problem in three steps. Step I includes a newly proposed segmentation network for mandible based on a unified algorithm combining U-Net and DenseNET with carefully designed network architecture parameters. In Step II, we propose a geodesic learning method to learn true and more accurate spatial relationship of landmarks on the mandible. Finally, in Step III, we identify each landmark's location by a classification framework where we utilize an LSTM learning algorithm. Mandible landmarks that we aim to locate are the following: Menton (Me), Condylar Left (Cd_L), Condylar Right (Cd_R) , Coronoid Left (Cor_L), Coronoid Right (Cor_R), Infradentale(Id), B point (B), Pogonion (Pg), and Gnathion (Gn). Positions of these landmarks are shown in Figure 5.1(b).

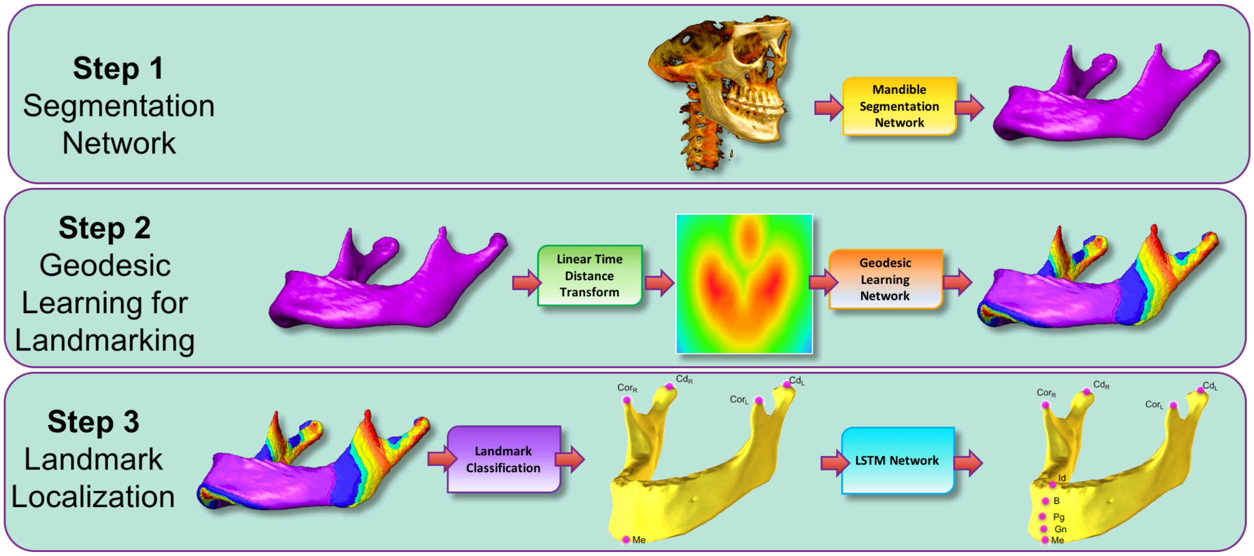
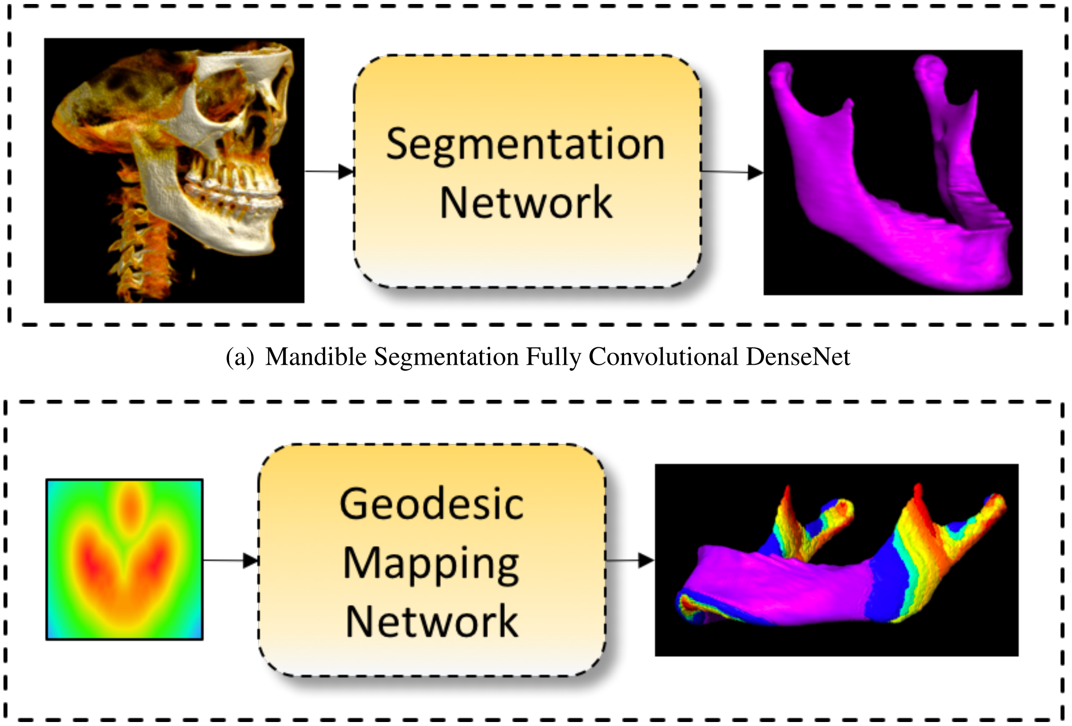


Figure 5.2: The framework implemented in this dissertation starts with a Fully Convolutional DenseNet for Mandible Segmentation. Following the Mandible Segmentation, Linear Time Distance Transform (LTDT) of the Mandible Bone is generated. Next, a U-NET is used for Mandibular Geodesic Learning which transforms 3D LTDT into combined 3D Geodesic Map of the mandibular landmarks Menton (Me), Condylar Left (Cd_L), Condylar Right (Cd_R), Coronoid Left (Cd_L), and Coronoid Right (Cd_R). After classification of 5 Mandibular Landmarks, a long short-term memory (LSTM) Network is used to detect Infradentale (Id), B point (B), Pogonion (Pg), and Gnathion (Gn) mandibular landmarks according to the detected position of the Menton (Me) landmark.

5.2 Step 1: Segmentation Network

CNN based approaches such as U-Net [60], fully convolutional network (FCN) [99], and encoder-decoder CNNs [100] have achieved increasing success in image segmentation. These methods share the same spirit of obtaining images at different resolutions by consecutive downsampling and upsampling to make pixel level predictions. Despite the significant progress made by such standard approaches towards segmentation, they often fail to converge in training when faced with objects with high variations in shape and/or texture, and complexities in the structure. Another challenge is the optimization of massive amount of hyper-parameters in deep nets. Inspired by the recently



(a) Mandible Segmentation Fully Convolutional DenseNet

(b) Geodesic Learning U-Net

Figure 5.3: (a) Mandibular bone segmentation framework (b) Geodesic Landmark Detection framework for landmarks Menton (Me), Condylar Left (Cd_L), Condylar Right (Cd_R) , Coronoid Left (Cor_L), and Coronoid Right (Cor_R), (Figure 5.1(b)).

introduced notion of densely connected networks (DenseNET) for object recognition [101], a new network architecture was presented by Jégou et al. [3] for semantic segmentation of natural images, called Fully Convolutional DenseNET (or *Tiramisu* in short). In this study, we adapted this Tiramisu network for medical image segmentation domain through significant modifications:

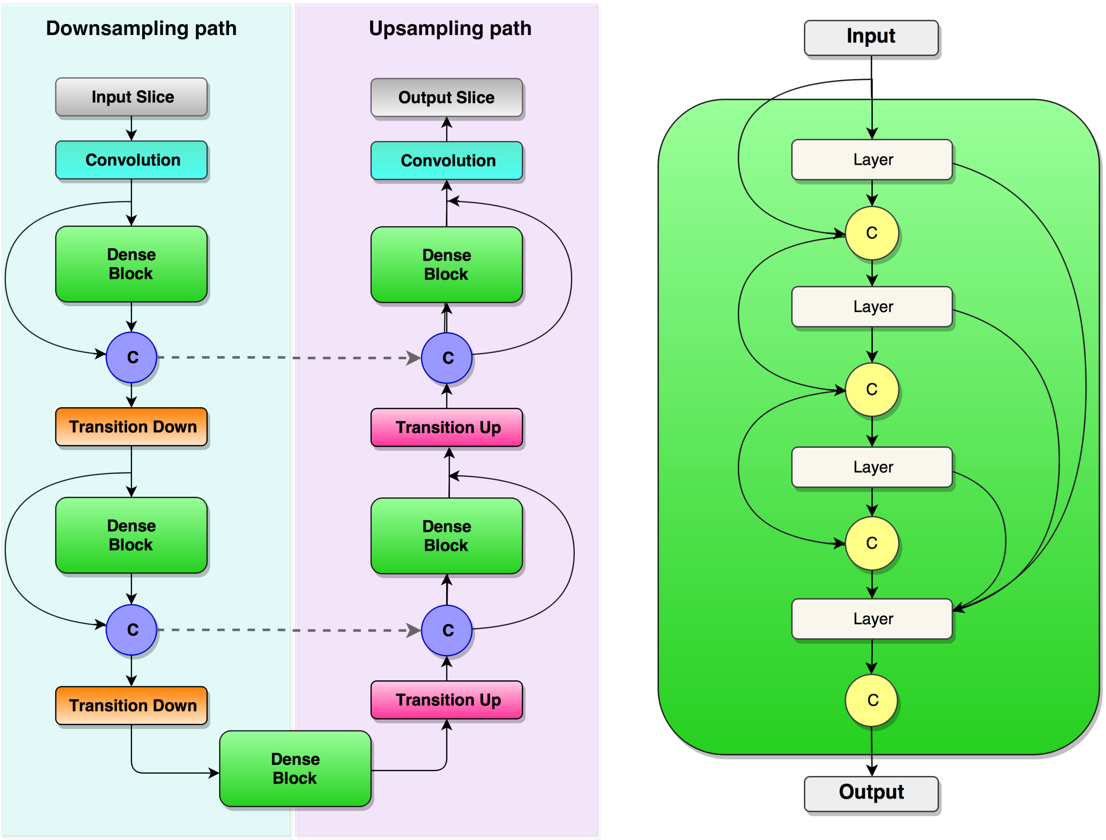
(1) We set all default pooling functions (often they are defined as max pooling) with average pooling to increase pixel-level predictions. Although pooling functions in the literature have been reported to perform similarly in various tasks, we hypothesized that average pooling was more suitable for pixel level predictions because average pooling identifies the extent of an object while max-pooling only identifies the discriminative part.

(2) We explored the role of dropout regularization on segmentation performance with respect to the commonly used batch normalization (BN) and pooling functions. Literature provides mixed evidence for the role of these regularizers.

(3) We investigated the effect of growth rate (of dense block(s)) on the segmentation performance. While a relatively small growth rate has been found successful in various computer vision tasks, the growth rate of dense blocks is often fixed and its optimal choice for segmentation task has not been explored yet.

(4) We examined appropriate regularization as well as network architecture parameters, including number of layers, to avoid the use of post-processing methods such as CRF (conditional random field). It is common in many CNN-based segmentation methods to use such algorithms so that the model predictions are further refined because the segmentation accuracy is below an expected range.

Figure 5 illustrates the Tiramisu network architecture (a) and the content of a dense block (b), respectively. Tiramisu network is extremely deep, including 103 layers [3] as compared to the U-Net which has only 19 layers in our implementation. The input of the Tiramisu network was the 2D sagittal slices of the CBCT scan of patients with CMF deformities, and the output was the binary 2D sagittal slices with the mandible segmented (see Figure 4 for an example workflow of a 2D slice). The architecture consisted of 11 dense blocks with 103 convolutional layers. Each dense block contained a variable length of layers and the growth rate was set specifically for each dense block based on extensive experimental results and comparison. The network was composed of approximately $9M$ trainable parameters. We trained the revised Tiramisu from scratch without the need for data augmentation and complex post-processing. Details of the network parameters are given in Tables 5.1 and 5.2.



(a) Fully Convolutional DenseNet with 103 layers.

(b) Content of a dense block.

Figure 5.4: (a) General architecture of the Tiramisu [3] is illustrated. The architecture is composed of downsampling and upsampling paths including Convolution, Dense Block, Concatenation (C), Skip Connection (dashed lines), Transition Down, and Transition Up layers. Concatenation layer appends the input of the dense block layer to the output of it. Skip connection copies the concatenated feature maps to the upsampling path. (b) A sample dense block with 4 layers is shown to its connections. With a growth rate of k, each layer in dense block appends k feature maps to the input. Hence, the output contains $4 \times k$ features maps.

5.3 Step 2: Geodesic Learning for Landmarking

We approach the problem of anatomical landmarking (landmark detection) as a learning problem. The state-of-the-art method in the literature adopts a U-Net architecture to learn the locations of the anatomical landmarks [4]. For a given 3D CBCT scan $\mathbf{X}$ and a landmark l, authors [4] created

Table 5.1: The network architecture of the Tiramisu segmentation engine.

Layers applied	**# of feature maps**
Input	1
3×3 Convolution	48
Dense Block (4 layers) + Transition Down	112
Dense Block (5 layers) + Transition Down	192
Dense Block (7 layers) + Transition Down	304
Dense Block (10 layers) + Transition Down	464
Dense Block (12 layers) + Transition Down	656
Dense Block (15 layers)	896
Transition Up + Dense Block (12 layers)	1088
Transition Up + Dense Block (10 layers)	816
Transition Up + Dense Block (7 layers)	578
Transition Up + Dense Block (5 layers)	384
Transition Up + Dense Block (4 layers)	256
1×1 Convolution	2
Softmax	2

three displacement maps $\mathbf{D}^{l,x}, \mathbf{D}^{l,y}, \mathbf{D}^{l,z}$ corresponding to x, y, and z axes [4]. That is, if there are N_l landmarks, $N_l \times 3$ displacement maps are generated. Displacement maps, also called *heatmaps*, were created using a simple Euclidean metric measuring the distance of a landmark to a reference point ((0,0) index of image). Although the method is simple to implement and efficient within the multi-task learning platform, it does not incorporate information about the object of interest (mandible) and works on the image space. In addition, the method generates a large number of heatmaps when the number of landmarks is high. Lastly, the method operates directly on the Euclidean space and it does not capture the underlying data distribution, which is non-Euclidean in nature.

To alleviate these problems and to solve the landmarking problem directly on the shape space, we propose to use a Geodesic Distance Transform to learn the relationship of landmarks directly on the shape space (mandible surface). To this end, we first apply linear time distance transform

Table 5.2: The network architecture parameters of the Tiramisu segmentation engine

Hyper-Parameters	**Value**
Learning-Rate	0.00005
Drop-out	0.2
Network Weight Initialization	Xavier Initializer
Bias Initializer	Zero Initializer
Activation Function	ReLu
Growth Rate	24
Normalization	Batch Normalization
Network Parameters	**Value**
Pooling	Average
Batch-Size	3
Optimization	Adam

(LTDT) [102] to the segmented mandible images (i.e., binary) and generate signed distance maps. Assuming $\mathbf{I}$ is a 3D-segmented binary image (mandible) obtained at Step 1 from a given CBCT scan $\mathbf{X}$ in the domain $\Omega = \{1, ..., n\} \times \{1, ..., m\}$, Mandible *M* is represented by all white voxels ($I(v) = 1$), while Mandible complement (background) M^C is represented by all black voxels ($I(v) = 0$) [103]:

$$\begin{aligned} \mathbf{M} &= \{v \in \Omega | I(v) = 1\}, \\ \mathbf{M^C} &= \{v \in \Omega | I(v) = 0\}. \end{aligned} \tag{5.1}$$

LTDT represents a map such that each voxel v is the smallest Euclidean distance from this voxel to the $\mathbf{M^C}$:

$$LTDT(v) = min\{dist(v, q) | q \in \mathbf{M^C}\}. \tag{5.2}$$

Then, the signed LTDT, namely sLTDT, of $\mathbf{I}$ for a voxel v can be represented as:

$$sLTDT(v) = \begin{cases} LTDT(v) & \text{if } v \in \mathbf{M}, \\ -min\{dist(v,q)|q \in \mathbf{M}\} & \text{if } v \in \mathbf{M^C}. \end{cases} \tag{5.3}$$

For each landmark l, we generate a geodesic distance map $\mathbf{D}_l^G$. To do so, we find the shortest distance between landmark l and each voxel v as:

$$\mathbf{D}_l^G(v) = \begin{cases} \min \pi(l,v) & \text{if } v \in \mathbf{M}, \\ \text{inf} & \text{if } v \in \mathbf{M^C}, \end{cases} \tag{5.4}$$

where π indicates all possible paths from the landmark l to the voxel $v(v \in \mathbf{M})$. Since the shortest distance between two points is found on the surface, it is called geodesic distance [104, 105] as a convention. To find the shortest path π, we applied Dijkstra's shortest path algorithm. For each landmark l, we generated one geodesic map as $\mathbf{D}_l^G$. For multiple landmarks, as is the case in our problem, we simply combined the geodesic maps to generate one final heatmap, which includes location information for all landmarks. Final geodesic map for all landmarks was obtained through hard minimum function $\mathbf{D}_{\mathbf{I}}^G = \min(\mathbf{D}_{l_1}^G \circ \mathbf{D}_{l_2}^G \circ ... \circ \mathbf{D}_{l_n}^G)$, where $\circ$ indicates pixel-wise comparison of all maps. In other words, the final geodesic map $\mathbf{D}_{\mathbf{I}}^G$ includes n extrema (minimum) identifying the locations of the n landmarks.

To learn the relationship of n landmark points on the mandible surface, we designed a landmark localization network, based on the Zhang's U-Net architecture [4]. Tiramisu network could perhaps be used for the same purpose. However, the data was simplified in landmark localization due to geodesic distance mapping, and Zhang's U-Net uses only 10% of the overall parameter space for landmark localization. The improved Zhang's U-Net accepts $2D$ slices of the signed distance transform of the segmented mandible ($\mathbf{I}$) as the input, and produces the $2D$ geodesic map ($\mathbf{D}_{\mathbf{I}}^{G}$) revealing the location of N_l landmarks as the output. The details of the landmark localization architecture (improved version of the Zhang's U-Net) with 19 layers and parameters are given in Tables 5.3 and 5.4, respectively. Briefly, the encoder path of the U-Net was composed of 3 levels. Each level consisted of (multiple) application(s) of convolutional nodes: 5×5 convolutions, batch normalization (BN), rectified linear unit (ReLU), and dropout. Between each level max pooling, downsampling with a stride of 2, was performed. Similar to the encoder path, the decoder path was also composed of 3 levels. In contrast to encoder path, dropout was not applied in the decoder path. Between the levels in the decoder path, upsampling operation was applied. To emphasize the high-resolution features that may be lost in the encoder path, copy operation was used in the decoder path. Copy operation, as the name implies, concatenated the features at the same $2D$ resolution levels from the encoder path to the decoder path.

We have chosen the optimization algorithm as RMSProp [106] due to its fast convergence and adaptive nature. The initial learning rate was set to 1e-3 with an exponential decay of 0.995 after each epoch (Table 5.4). At the end of the decoder path, softmax cross entropy was applied as a loss function because mean squared error (MSE) caused serious convergence issues. We quantized the geodesic map in the range $[0 - 20]$, where the limit 20 was set empirically. The network was composed of $\approx 1M$ trainable parameters. Compared to the Zhang's U-Net [4], in our improved implementation, in addition to the 5×5 convolutions, on the expanding path at level 2, we kept the symmetry in the number of features obtained as in the contracting path. These alterations made sure Zhang's U-Net to work without failures.

Table 5.3: The network architecture of the improved Zhang's U-Net for sparsely-spaced landmarks

Layers applied	**Slice Size**	**Number of feature maps**
Input	256×256	1
5×5 Convolution	256×256	32
5×5 Convolution	256×256	32
Max-pooling	128×128	32
5×5 Convolution	128×128	64
5×5 Convolution	128×128	64
Max-pooling	64×64	64
5×5 Convolution	64×64	128
5×5 Deconvolution	64×64	64
Upsampling + Copy	128×128	128
5×5 Deconvolution	128×128	64
5×5 Deconvolution	128×128	32
Upsampling + Copy	256×256	64
5×5 Deconvolution	256×256	32
5×5 Deconvolution	256×256	32
5×5 Deconvolution	256×256	21
Softmax	256×256	21

5.4 Step 3: Localization of Closely-Spaced Landmarks

Fusion of geodesic maps through pixel-wise hard-coded minimum function is reliable when landmarks are sufficiently distant from each other. In other words, if landmarks are very close to each other, then the combined geodesic map $\mathbf{D}_{\mathbf{I}}^{G}$ may have instabilities in locating its extrema points. In particular for our case, it was not possible to localize specifically "Menton" and other mid-sagittal closely-spaced landmarks in a clinically acceptable error range (i.e., $\leqslant 3mm$). In order to avoid such scenarios, we divided the landmarking process into two distinct cases: learning closely-spaced and sparsely-spaced landmarks separately. First, we classified the mandible landmarks into

Table 5.4: The network architecture parameters of the improved Zhang's U-Net for sparsely-spaced landmarks

Hyper-Parameters	**Value**
Learning-Rate	1e-3
Decay-Rate	0.995
Drop-out	0.2
Network Weight Initialization	Xavier Initializer
Bias Initializer	Zero Initializer
Normalization	Batch Normalization
Pooling	Maxpool
Batch-Size	3
Optimization	RMSProp

sparsely and closely-spaced sets. Sparsely-spaced landmarks (N=5) were defined in the inferior, superior-posterior-left, superior-posterior-right, superior-anterior-left, and superior-anterior-right regions. Closely-spaced landmarks (N=4) were defined as the ones that were closely tied together (Infradentale (Id), B point (B), Pogonion (Pg), and Gnathion (Gn)).

Note that these anatomical landmarks often reside on the same sagittal plane in the same order according to the mid-point of the lower-jaw incisors. We propose to capture this order dependence by using an LSTM architecture in the sagittal axis of the images containing the landmark "Menton" (Figure 5.5). The rationale behind this choice was that LSTM network is a type of RNN introduced by Hochreiter et al. [107] in 1997, modeling the temporal information of the data effectively. Although the imaging data that we used does not include temporal information in the standard meaning, we modeled the landmark relationship as a temporal information due to their close positioning in the same plane. This phenomenon is illustrated in Figure 5.5. The input data to the LSTM network was a 64×64 mandible binary boundary image of the sagittal plane of the landmark Me, and the output was a vector of 0's and 1's: while 0 refers to non-landmark location, 1 refers to a landmark location in the sagittal axis. Figure 5.6 shows further details of the LSTM

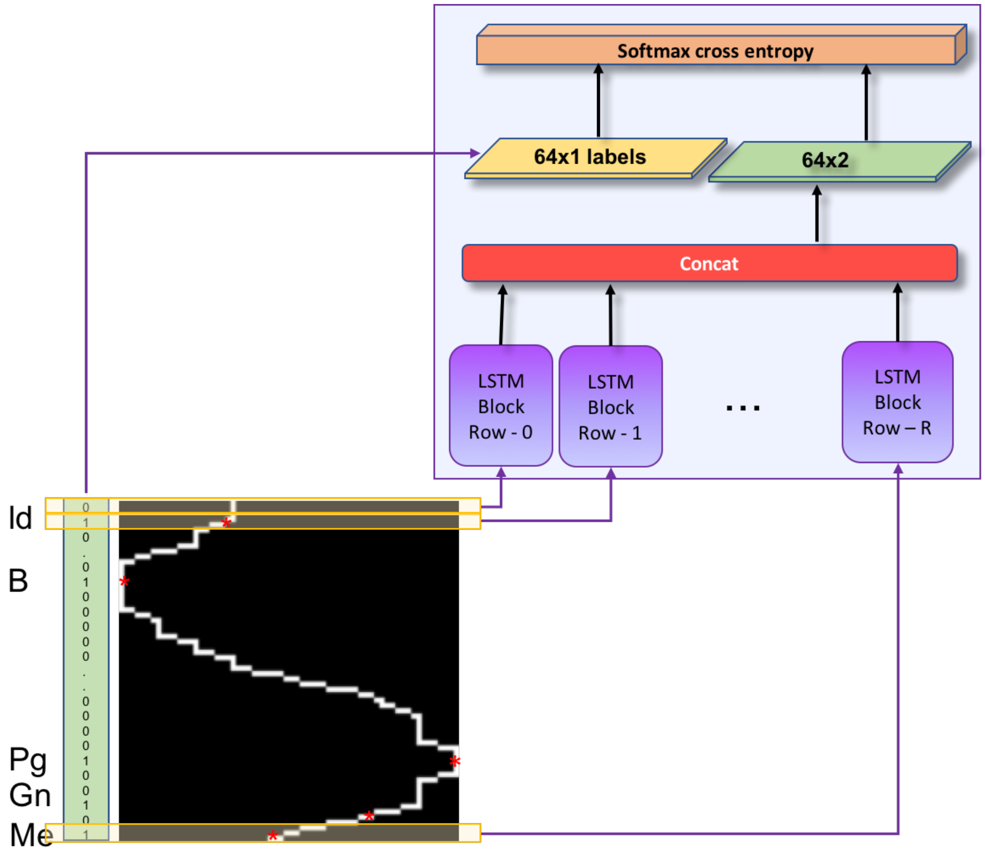
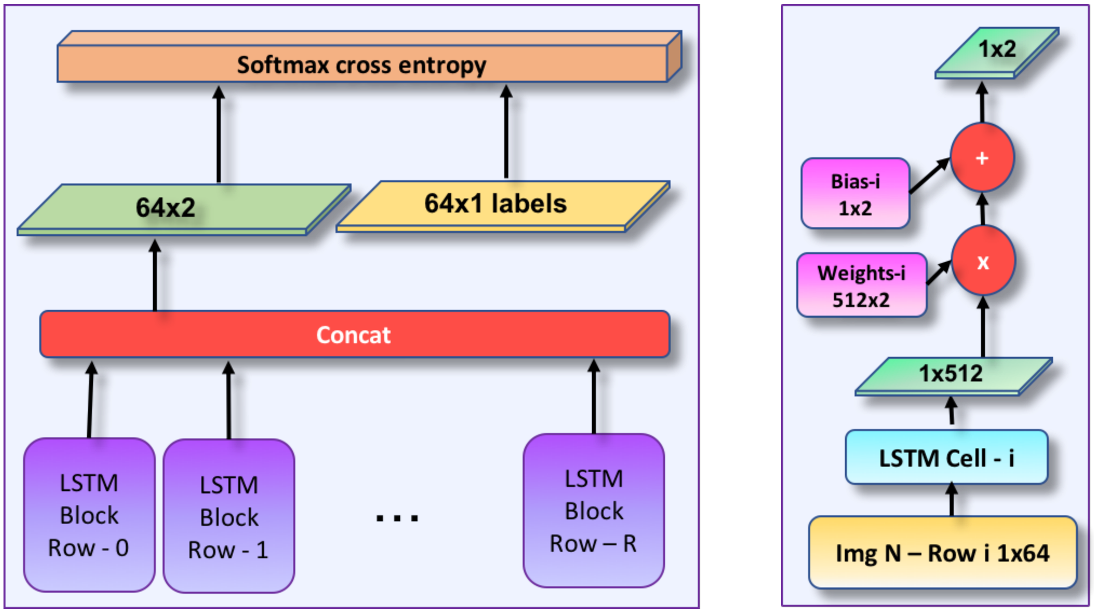


Figure 5.5: LSTM network input-output. Each row of the scaled sagittal boundary image is input to the corresponding LSTM block, and binary $1D$ vector of locations annotated as landmark (1), or no-landmark (0) is output.

network and content of a sample LSTM block that we used for effective learning of closely-spaced landmarks.

To generate the training data, the sagittal slice containing the closely-spaced landmarks “Menton”, “Gnathion”, “Pogonion”, “B-point” and “Infradentale” was scaled into a binary boundary image of size 64×64. The 5 landmark locations (marked by red circles in Figure 5.5) on this boundary image were parameterized as $(\boldsymbol{x},\boldsymbol{y})$, where $\boldsymbol{y}$ is the row number in the range 0 to 64, and $\boldsymbol{x}$ is the white boundary column number of the corresponding row $\boldsymbol{y}$.



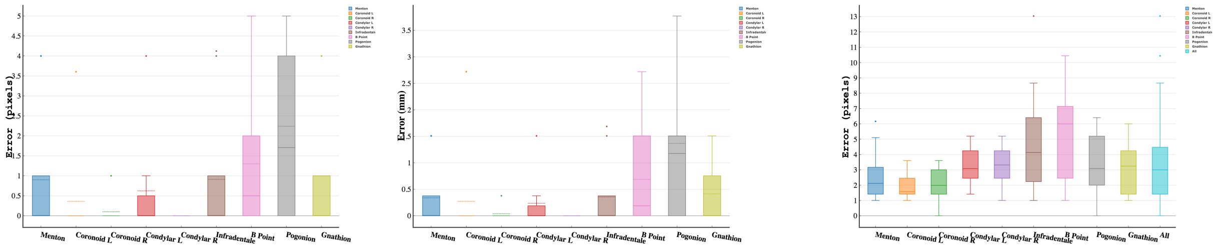
(a) LSTM Network (b) LSTM Block

Figure 5.6: Details of the network architecture (LSTM) for identifying closely-spaced landmarks. Gnathion (Gn), Pogonion (Pg), B Point (B), and Infradentale (Id) are determined once the Menton (Me) is detected through U-Net architecture as shown in Step 3 of the Figure 5.2. Input image resolution is RxK, and the LSTM cell is composed of 512 hidden units.

LSTM network was composed of 64 cells (Figure 5.6), and each cell in the LSTM network consisted of 512 units. The training images were row-wise input to the LSTM network such that n^{th} row was input to the corresponding n^{th} cell of the network. The output of each cell was multiplied by 512×2 weight and 1×2 bias was added. The resultant 1×2 tensors at each cell were concatenated and softmax cross entropy was applied as a loss function.

5.5 Training Framework: End-to-end vs. Sequential vs Mixed

Since the proposed learning system is complex, it is worth to explore whether gradient-descent learning system can be applied to the system as a whole (called *end-to-end*). For this purpose,



(a) Pixel Space Errors

(b) Volume Space Errors

(c) Expert Reading Variations

Figure 5.7: (a) Errors in pixel space, (b) errors in the volume space, (c) inter-observer reading variations in pixel space.

first, we evaluated the performances of each network individually, so named *sequential training* followed by an engineering approach for concatenation of the three networks. Since end-to-end learning systems require all modules of the complex system to be differentiable, our proposed system was not fully eligible for this learning type. It is because the 3^{rd} module (LSTM network for closely-spaced landmark localization) had differentiability issues for the given loss function. Therefore, we trained the first and second modules in an *end-to-end* manner while integrating the third network module into this system sequentially. In summary, we devised two alternative methods to solve our overall goal: in the first solution, the overall system was considered as a sequential system. In the second solution, the first two modules of the system were trained in an end-to-end manner with the inclusion of the third module as a sequential block. Owing to the usage of sequential and end-to-end frameworks together, we named the second solution as “mixed”.

Although end-to-end networks are conceptually and mathematically beautiful, it has a strict condition that each module should be differentiable with respect to the loss function so that a positive impact can be obtained on the final objective. However, as stated in [108] and [109], when some modules are not differentiable (as the third module of our proposed method), or when the system is too complex with sparse modules, the overall results may be inferior compared to the sequential method. Due to the differentiability issue in the third module, our system falls into this category. That is, the input to the 3^{rd} module was the $2D$ sagittal slice containing the anatomical landmark "Menton". Since not every output slice in the training could be used for the 3^{rd} module, differentiability was lost. In addition, we observed that unless the first two modules were close to the converging state, it was not possible to locate "Menton" more precisely than a random guess. Due to the requirement of convergence within this module, eventually, it was not possible to apply LSTM training in a truly end-to-end manner.

5.6 Experiments and Results

5.6.1 Data description:

Anonymized CBCT scans of 50 patients (30 female and 20 male, mean age = 22.4 years, standard deviation = 9.6 years) were included in our analysis through an IRB-approved protocol and data sharing agreement between UCF and NIH. These patients had craniofacial congenital birth defects, developmental growth anomalies, trauma to the CMF, surgical intervention, and included pediatric and adult patients. All images were obtained on a CB MercuRay CBCT system (Hitachi Medical Corporation, Tokyo, Japan). The 12-inch field of view was required for this study to capture the entire length of the airway and was scanned at 10 mA and 100 Kvp. The equivalent radiation dosage for each scan was approximately 300 mSv. After the study had begun, the machine was modified to accommodate 2 mA for the same 12-inch field of view, thus lowering the equivalent radiation dosage for each scan to approximately 132.3 mSv. Each patient's scan was re-sampled from $512 \times 512 \times 512$ to $256 \times 256 \times 512$ to reduce computational cost. In-plane resolution of the scans was noted either as $0.754mm \times 0.754mm \times 0.377mm$ or $0.584mm \times 0.584mm \times 0.292mm$. Apart from highly diverse nature of this data set, the following image-based variations have also been confirmed: aliasing artifacts due to braces, metal alloy surgical implants (screws and plates), dental fillings, and missing bones or teeth.

Additionally, we tested and evaluated our algorithm(s) using the MICCAI Head-Neck Challenge 2015 dataset [110]. MICCAI Head-Neck Challenge 2015 dataset was composed of manually annotated CT scans of 48 patients from the Radiation Therapy Oncology Group (RTOG) 0522 study (a multi-institutional clinical trial [111]). For all data, the reconstruction matrix was 512×512 pixels. The in-plane pixel spacing was isotropic, and varied between $0.76mm \times 0.76mm$ and $1.27mm \times 1.27mm$. The range of the number of slices of the scans were 110-190. The spacing in the z-direction was between $1.25mm$ and $3mm$ [110]. In the challenge, there were three test results provided, where test data part 1 (off-site data) and part 2 (on-site data) did not have publicly available manual annotations to compare to our performances. Hence, we compared our test results to the the cross-validation results as provided in [112].

Training deep networks: We have trained our deep networks with 50 patients' volumetric CBCT scans in a 5-fold cross validation experimental design. Since each patient's scan includes 512 slices (i.e., $2D$ images with 256×256 pixels in-plane), we had a total of $25,600$ images to train and test. In each training experiment, we have used $20,480$ $2D$ images to train the network while the remaining slices $(5,120)$ were used for testing. This procedure was repeated for each fold of the data, and average of the overall scores were presented in the following subsections.

5.6.2 Evaluation metrics and annotations:

Three expert interpreters annotated the data (one from the NIH team, two from the UCF team) independently. Inter-observer agreement values were computed based on these three annotations. Later, second and third experts (from the UCF team) repeated their manual annotations (after one month period of their initial annotations) for intra-observer evaluations. Experts used freely available 3D Slicer software for the annotations. Annotated landmarks were saved in the same format of the original images, where landmark positions in a neighborhood of $3 \times 3 \times 3$ were marked according to the landmark ID while the background pixels were marked as 0.

A simple median filtering was used to minimize noise in the scans. No other particular preprocessing algorithm was used. Experiments were performed through a 5-fold cross-validation method. Intersection of Union (IoU) metric was used to evaluate object detection performance. For evaluating segmentation, we used the standard DSC (dice similarity coefficient), Sensitivity, Specificity, and HD (Hausdorff Distance) (100% percentile). As a convention, high DSC, sensitivity, specificity and low HD indicate a good performance. The accuracy of the landmark localization was evaluated using the detection error in pixel space within a $3 \times 3 \times 3$ bounding box. Inter-observer agreement rate was found to be 91.69% for segmentation (via DSC).

5.6.3 Evaluation of Segmentation

The proposed segmentation framework achieved highly accurate segmentation results despite the large variations in the imaging data due to severe CMF deformities. Table 5.5 summarizes the segmentation evaluation metrics and number of parameters used for the proposed and the compared networks. The proposed segmentation network outperformed the state-of-the-art U-Net [60]. Specifically, we have improved the success of the baseline U-Net framework by increasing the number of layers into 19. In terms of the dice similarity metric, both improved Zhang's U-Net and the proposed segmentation network were statistically significantly better than the baseline U-Net ($P = 0.02$, t-test). In summary, (i) there is no statistically significant difference noted between our proposed method and the manual segmentation method ($P = 0.77$); (ii) there is a statistically significant difference between our proposed method and the baseline U-Net ($P = 0.02$); (iii) there is no statistically significant difference noted between the proposed method and our improvement over the Zhangs U-Net ($P = 0.28$). It is also worth to note that the proposed Tiramisu network performed more robustly in training, converging faster than the improved Zhang's U-Net despite the larger number of parameters in the Tiramisu.

Table 5.5: Evaluation of the segmentation algorithms. Higher IoU(%) and DSC (%), and lower HD (mm) indicate better segmentation performance. Improved Zhang's U-Net is built on top of Zhang's U-Net implementation [4].

Method	IoU	DSC	HD	Layers	# of params.
Baseline U-Net [60]	100	91.93	5.27	31	≈ 50M
Improved Zhang's U-Net	100	93.07	5.87	19	≈ 1M
Proposed (Tiramisu)	100	**93.82**	**5.47**	103	≈ 9M

We evaluated the segmentation performances on different datasets and training styles (sequential vs. mixed learning) and summarized the results in Table 5.6. With the MICCAI Head-Neck Challenge 2015 dataset, we obtained a dice accuracy of 93.86% compared to 90% [112]. High accuracies of the MICCAI Head-Neck Challenge 2015 and the NIH datasets imply the robustness of the Tiramisu segmentation network. It should be noted that MICCAI Head-Neck Challenge 2015 dataset contains mainly scans with imaging artifacts as well as different diseases. Closer inspection of Table 5.6 also shows that a simple post-processing step such as "Connected Component Analysis" and "3D fill" were important to decrease the number of the false positives and false negatives in the challenge dataset. The slightly lower performances of mixed training with Tiramisu network for both segmentation and landmark localization can be explained by the increased number of parameters but insufficient dataset size to derive learning procedure as a whole. Sequential learning was sufficient to obtain good results in segmentation, though.

5.6.4 Evaluation of Landmark Localization

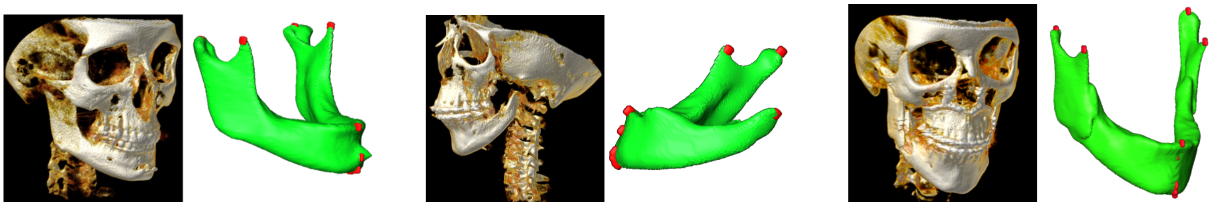
Ground truth annotations (manual segmentation and anatomical landmarking) were performed by three experts independently. Inter-observer agreement rate was 91.69%. Figure 5.7(c) presents per

Table 5.6: Segmentation performances in different datasets, training paradigms (mixed vs. sequential), and post-processing algorithms.

	Post-processing	DSC(%)	Sensitivity(%)	Specificity(%)	HD(mm)
Sequential Tiramisu Segmentation MICCAI 2015	–	92.30	86.43	99.96	5.09
	connected component analysis, $3D$ fill	93.86	95.23	99.99	4.58
Sequential Tiramisu Segmentation NIH Dataset	–	92.61	93.42	99.97	8.80
	connected component analysis, $3D$ fill	93.82	93.42	99.97	6.36
Mixed Tiramisu Segmentation → U-Net Landmark Localization NIH Dataset	–	92.09	92.10	99.96	8.30
	connected component analysis, $3D$ fill	92.28	92.10	99.96	7.11
Mixed Tiramisu Segmentation → Tiramisu Landmark Localization NIH Dataset	–	90.10	90.53	99.97	8.80
	Connected component analysis, $3D$ fill	90.10	90.52	99.97	6.36

landmark and overall expert reading variations of landmarking in the pixel space. We observed that there was an average 3 pixel errors among the experts. Hence, any landmarking algorithm leading to error within 3 pixel range can be considered a clinically acceptable level of success. Figures 5.7(a) and 5.7(b) summarize the proposed algorithm's landmark localization errors in the pixel space and the volume space, respectively.

The mean and median volume space errors for each landmark are presented at Table 5.7. The errors in the pixel space (Figure 5.7(a)) were less than 3 pixels for all 9 landmarks, indicating that our method is highly accurate and can be used for clinical applications as it results in less variations than the inter-observer variation rate as explained earlier (Figure 5.7(c)).



(a) Patient-1

(b) Patient-2

(c) Patient-3

Figure 5.8: Experimental renderings demonstrating segmentation and landmark localization results of patients with high anatomical variability due to deformities and surgical intervention a) Genioplasty/chin advancement (43 yo), b) Absent left condyle-ramus unit (15 yo), c) Mandibular implant (17 yo).

Figure 5.8 presents three experimental results when there was high morphological variation and deformity. In Figure 5.8(a), due to the genioplasty with chin advancement and rigid fixation, there is a protuberance on the mandible distorting the normal anatomy. In Figure 5.8(b), condyle-ramus unit is absent on the left side of the mandible due to a congenital birth defect. The Geodesic Landmark Localization network successfully detected 4 landmarks. Note that the fifth landmark was on the missing bone, and it was not located as an outcome of the landmarking process. This is one of the strengths of the proposed method. In Figure 5.8(c), the patient had bilateral surgical implants along the ascending ramus (bicortical positional screws), and bilateral condyle and coronoid processes are fixed with these implants. The landmarking process was successful in this challenging case too.

We also evaluated the impact of segmentation accuracy on the landmark localization error (Figure 5.9). In this evaluation, we first grouped the testing scans into 2 groups according to their dice values as lower and higher segmentation accuracies (i.e., $\leq 90\%$ as lower, $> 90\%$ as higher). Next, we compared the landmark localization errors in pixel space for these two groups. In Figure 5.9, the landmarking process was robust to changes in segmentation accuracy, and never reached more than 3 pixels errors. It should also be noted that the mean and median segmentation accuracy were

Table 5.7: Landmark localization performances are evaluated for each anatomical landmark and with respect to different pooling functions. Errors (in mm) are given both in average (avg) and median (md) values.

		max pool	avg pool	stoc. pool	max pool + wo drop out
Me	avg	0.33	1.35	0.37	0.03
	md	0	0	0	0
Cor_L	avg	0.27	0.07	0	0
	md	0	0	0	0
Cor_R	avg	0.03	0.3	0.37	0.45
	md	0	0	0	0
Cd_L	avg	1.01	0.037	0.56	0.33
	md	0	0	0	0
Cd_R	avg	0	0.11	0.07	0.07
	md	0	0	0	0
Gn	avg	0.41	1.64	1.35	0.49
	md	0	0	0.18	0
Pg	avg	1.36	2.34	2.4	1.54
	md	1.17	0.75	1.6	0.75
B	avg	0.68	1.47	1.24	0.33
	md	0.18	0	0.56	0
Id	avg	0.35	1.74	0.75	0.52
	md	0	1.131	1.67	0

still very high in our experiments, leading to successful landmark localization even at the low end of the dice values. Overall, the landmark localization was robust to the segmentation step and a potential (visible) error may happen only when the Menton (closely-spaced landmark) is located incorrectly due to a potential segmentation error.

Table 5.7 summarizes the average and median errors of localized landmarks in millimeters with respect to different regularization methods, in particular pooling strategies. We observed that max pooling consistently outperformed other regularization methods. Unlike the segmentation problem, where average pooling was most effective in pixel level predictions, landmarking was driven by discriminative features, enhanced by max pool operation. All average and median errors of the

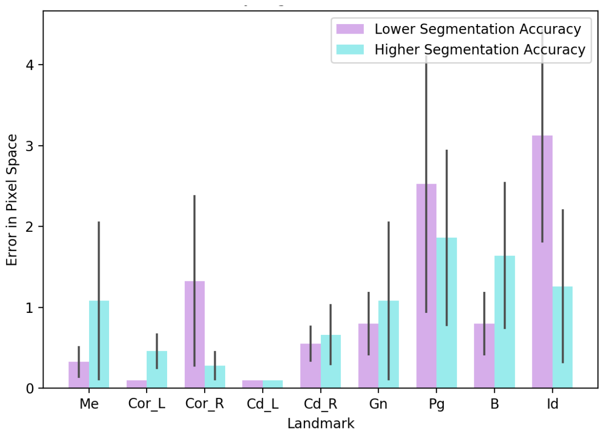


Figure 5.9: Impact of segmentation accuracy on the landmark localization process.

landmark localizations were within the clinically acceptable limits (less than 3 pixels).

5.6.5 Evaluation of the Segmentation Network Parameters

Table 5.8: Resulting segmentation DSC accuracies with respect to the drop ratio (avg pooling, ReLU, and growth rate of 24). Note that drop ratio of 0 denotes "no" use of dropout layer.

Drop Ratio	0.5	0.3	0.2	0.1	0.0
DSC(%)	91.21	93.37	**93.82**	92.90	92.88

5.6.5.1 Effect of pooling functions

After extensive experimental comparisons, we found that average pooling acts as a robust regularizer compared to other pooling functions such as max pooling and stochastic pooling (Table 5.8).

5.6.5.2 Disharmony between BN and dropout

We found that when BN is used in the network for segmentation purpose, the use of dropout is often detrimental except for only a drop rate of 20%. Similarly, we found that average pooling was the most robust pooling function compared to others when BN and dropout were used together.

Table 5.9: Effect of different growth rates on segmentation performance using Tiramisu with avg. pooling.

Growth Rate (k)	12	16	24	32
DSC(%)	92.63	93.36	**93.82**	92.60
HD(mm)	6.44	5.50	5.47	**5.02**

5.6.5.3 The role of growth rate in dense blocks

Tiramisu network with 103 layers (growth rate of 16) has a proven success in the computer vision tasks. In our experiments, we observed that a Tiramisu network with a growth rate of 24 and drop rate of 0.2 produces the best accuracies instead of growth rate of 16 as in computer vision tasks (See Table 5.9). Further, when no dropout is used (drop rate is 0), the growth rate performance inverses (See Table 5.10), implying the regularizing impact of employing dropout on the neural networks.

Table 5.10: Comparison of segmentation accuracies with respect to different regularization choices. Drop ratio of 0 denotes "no" use of dropout layer.

Pooling		max pool	max pool	avg pool	avg pool	stoc. pool	stoc. pool	avg pool	avg pool
Activation		ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	SWISH	SWISH
Growth Rate		16	24	16	24	16	24	16	24
DSC(%)	drop ratio=0.0	93.09	92.64	93.16	92.16	92.59	90.93	93.14	92.60
	drop ratio=0.2	93.10	93.08	93.36	**93.82**	92.14	92.53	91.79	93.67

5.6.6 Qualitative Evaluation

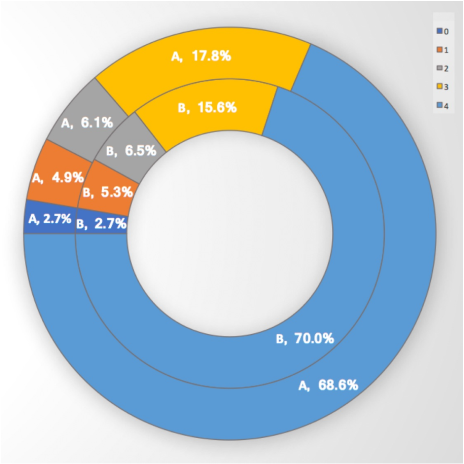


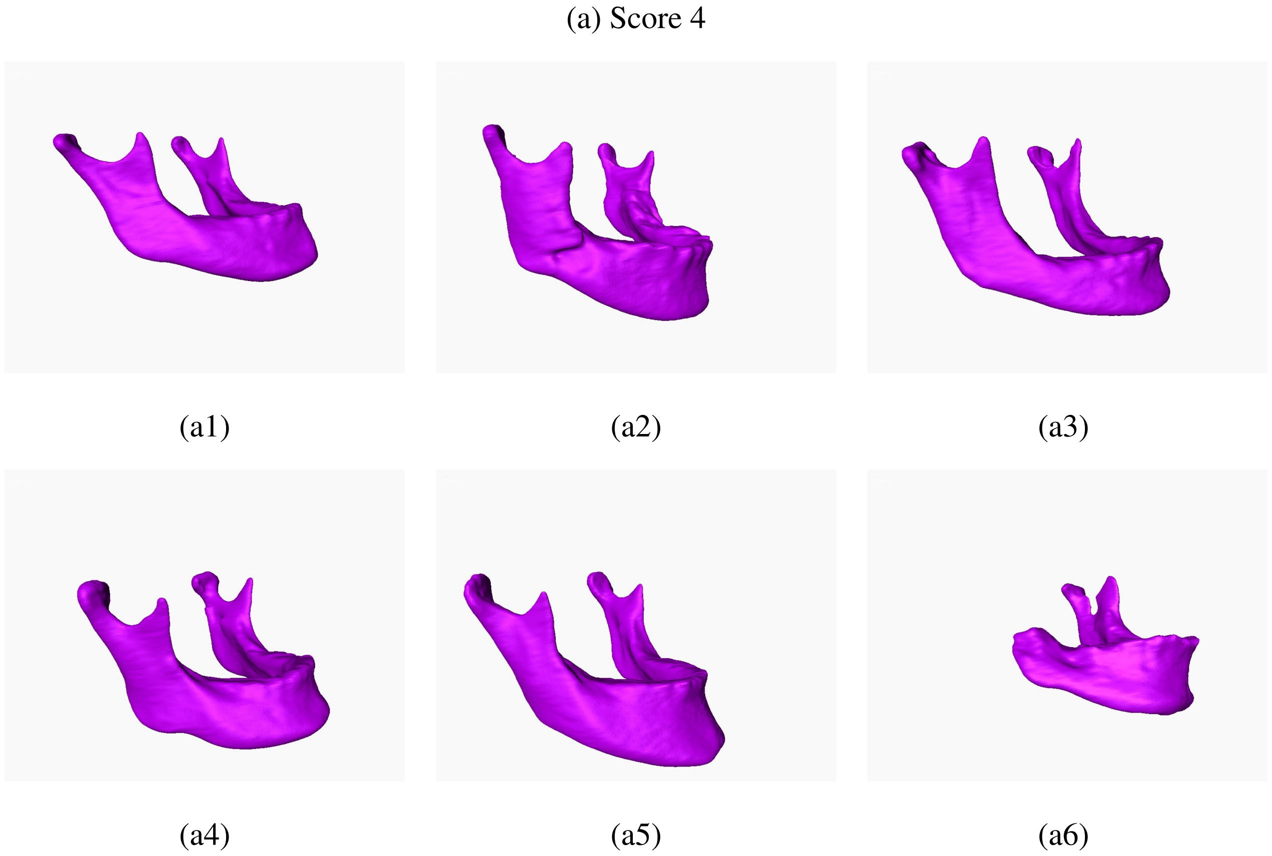
Figure 5.10: Summary of qualitative evaluation of 250 scans from the NIDCR/NIH dataset evaluated by 2 experts, **A** and **B**

.

5.6.6.1 The choice of activation functions

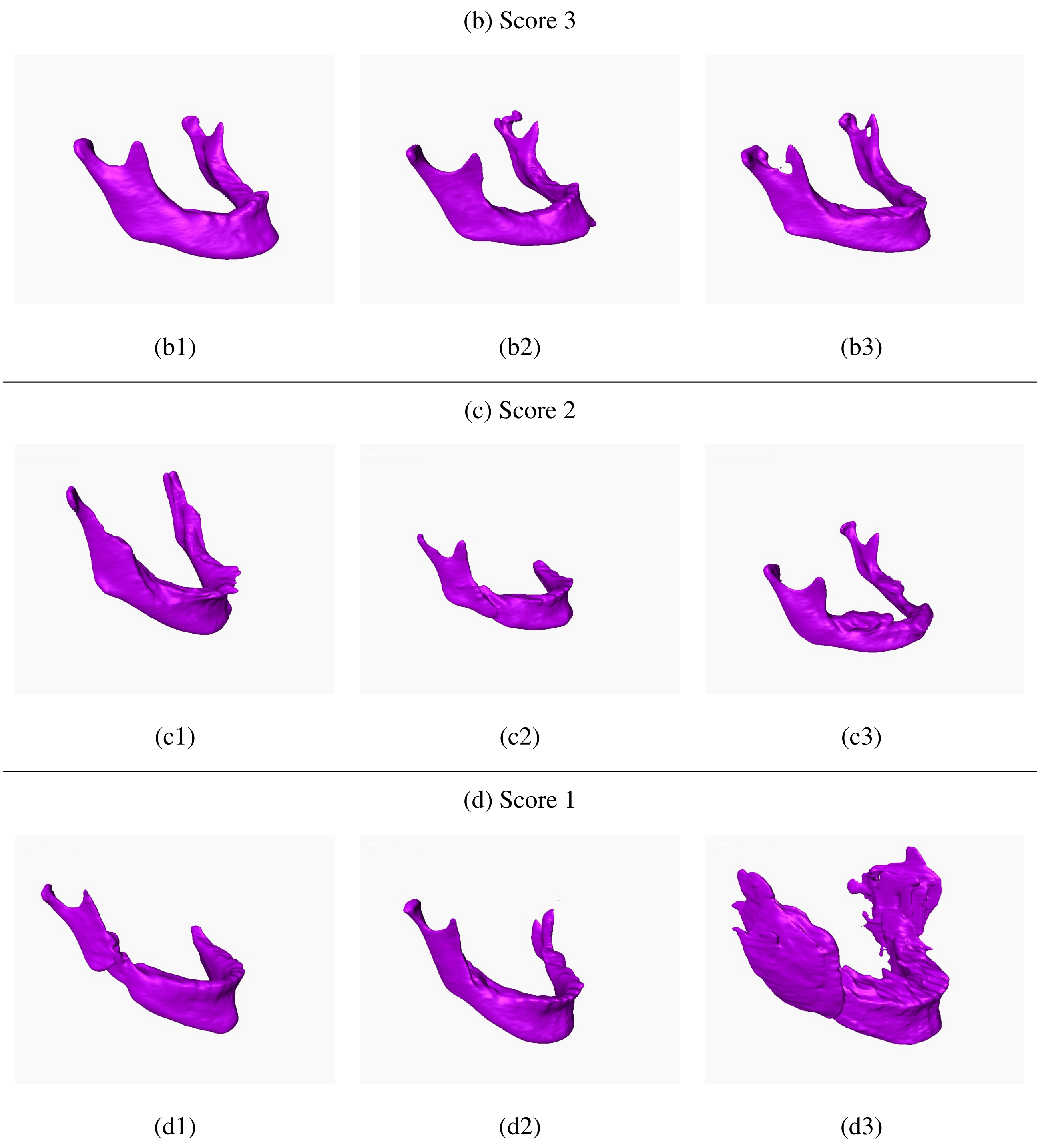
Although there have been many hand-designed activation functions proposed for deep networks, ReLU (rectified linear unit) became an almost standard choice for most CNNs. The main reason is due to its significant effect on the training dynamics and high task performances. More recently, another activation function, called "Swish" [113], was proposed. Unlike other activation functions, Swish was automatically determined based on a combination of exhaustive and reinforcement learning-based search. Authors showed that Swish tend to perform better than ReLU for very deep models. Since the proposed Tiramisu has 103 layers, we replaced all ReLU functions with Swish, which is a weighted sigmoid function $f(x) = x.sigmoid(\beta x)$, and explored the network behaviors. We summarized the network performance in Table 5.10. Overall, we did not observe significant differences between ReLU and Swish, but ReLU led into slightly better results in all sub-experiments.

Our total CBCT dataset is composed of 250 patient CBCT images provided by our collaborators at the NIDCR/NIH. Only 50 of them were manually annotated by the three experts. To measure the performance of the algorithm on all available scans, by following the routine radiologic evaluation of the scans, two experts visually scored the segmentation results in the range from 0 to 4, where 1 is unacceptable, 2 is borderline, 3 is acceptable at clinical level, and 4 is superior (excellent) (Figures 5.10 and 5.11). When the scan is completely distorted or mandible does not exist in its entirety in the scan, it is not possible to automatically segment mandible, hence a score of 0 is given.


(a) Score 4

(a1) (a2) (a3)

(a4) (a5) (a6)



(b) Score 3

(b1) (b2) (b3)

(c) Score 2

(c1) (c2) (c3)

(d) Score 1

(d1) (d2) (d3)

Figure 5.11: Qualitative Evaluation Scores of the Segmentation Results. The experts visually evaluated the performance of the segmentation of the 250 patient scans in the score range 1 to 4, where 1 is inferior. Examples of scans with scores 1-4 are presented.

Scores of 3 and 4 represent clinically acceptable segmentation, where score 3 may correspond to minor deformations in the segmentation. The left top part of the Mandible (Figure 5.11-a6) was missing, for instance, but it was still precisely segmented. The mandible (Figure 5.11-c2), for another example, was composed of two separate parts, and the algorithm detected the larger portion of the mandible. Further analysis showed that the scans that were scored as 1 or 2 were typically the ones with serious anatomical deformations. Approximately 5% of the segmentation results were scored as 1 by both experts **A** and **B** (Figure 5.10).

5.7 Discussion and Conclusion

Overall, the proposed networks (Tiramisu and improved Zhang's U-Net) have enjoyed fast convergence (around 20 epochs) and high accuracy in a very challenging CBCT dataset. Tiramisu was observed to have better converging and training ability compared to improved Zhang's U-Net. For landmark localization, improved Zhang's U-Net in the geodesic space has performed comparably to the validated operator manual landmarking (e.g., median displacement error of 0 mm in most landmarks). We also addressed some of the poorly understood concepts in deep network architecture (particularly designed for medical image analysis applications) such as the use of dropout and pooling functions for regularization, activation functions for modeling non-linearity, and growth rate for information flow in densely connected layers (See Chapter 5.6.5).

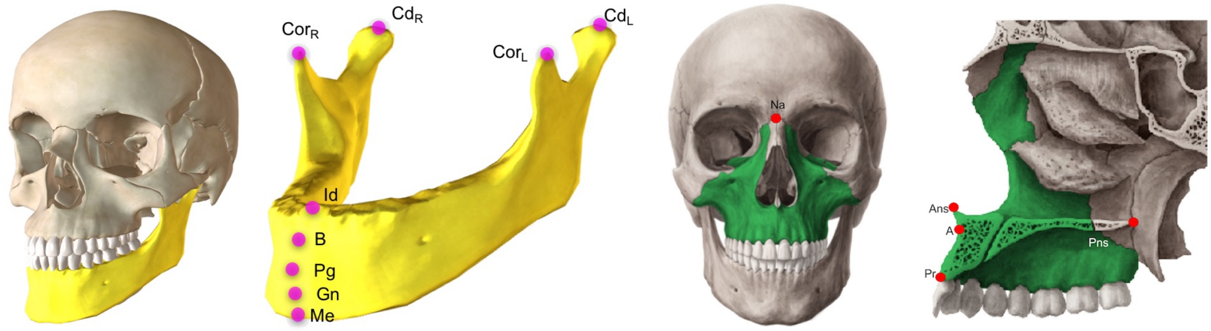
Fully convolution network (FCN) [99] has significantly changed the landscape of semantic image segmentation frameworks. Based on the FCN, Ronneberger et al. [60] introduced the U-Net which became the baseline for the current medical image segmentation tasks. The literature for particular medical image segmentation applications based on U-Net is vast; employing the encoder-decoder structure, dense connections, skip connections, residual blocks, and other types of architectural additions to improve segmentation accuracy for particular medical imaging applications. One major drawback of the U-Net framework is the inefficiency introduced by the significantly higher number of parameters to learn [114]. Hence, there is an anticipation for improvements in the efficiency and robustness of the U-Net type of architecture in the medical imaging field in the near future. One example of such studies, called Capsules [114], may be a good future alternative to what we propose herein.

In our study, we have focused on individual aspects of segmentation and landmarking, and have proposed novel architectural designs to address problems in both processes that have not been corrected in currently available systems. The natural extension of our work will be to formulate segmentation and landmarking problem within the multi-task learning algorithm, similar to the one proposed by Zhang et al. [4].

There are some limitations to our proposed method. Due to extensive memory and hardware requirement, we used pseudo-3D image analysis instead of fully 3D. A possible extension of our study will be to work on completely 3D space once hardware and memory supports are available. Another limitation of our work is to have a two-cascaded system for landmark localization instead of one because the hard-coded minimum function that we used for combining geodesic distances created additional artificial landmarks between those closely distributed landmarks. To overcome this problem, we showed a practical and novel use of LSTM-based algorithm to learn the locations of closely-spaced landmarks and avoided such problems. Exploration of different functions other than hard-coded minimum for closely-spaced landmark localization is subject to further theoretical investigation in geodesic distance maps.

A more radical approach to handle the two-cascaded system is anatomical landmarking without performing any segmentation. Although, CMF landmarks always reside on the boundaries of the CMF bones, and segmentation provides a very valuable input in this respect, in Chapter 6 we further explored the detection of the anatomical landmarks without employing segmentation.

CHAPTER 6: RELATIONAL REASONING NETWORK (RRN) FOR ANATOMICAL LANDMARKING

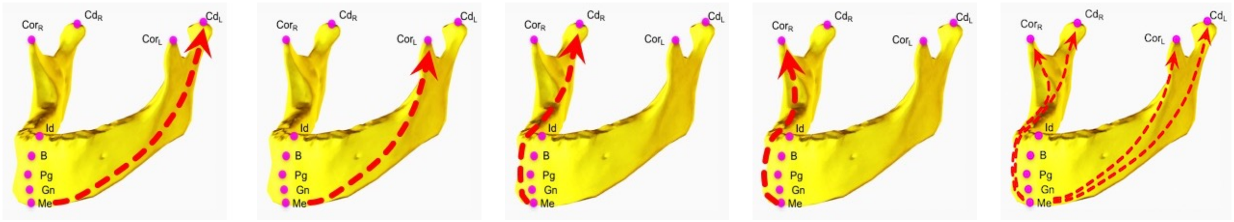


(a) Mandible [115]

(b) Maxilla [116]

Figure 6.1: Mandible and Maxilla anatomies, a) Mandibular Landmarks: Menton (Me), Condylar Left (Cd_L), Condylar Right (Cd_R) , Coronoid Left (Cor_L), Coronoid Right (Cor_R), Infradentale(Id), B point (B), Pogonion (Pg), and Gnathion (Gn), b) Maxillary Landmarks: Anterior Nasal Spine (ANS), Posterior Nasal Spine (PNS), A-Point (A), and Prostion (Pr), Nasal Bones Landmark: Nasion (Na).

Accurately identifying anatomical landmarks is a crucial step in the deformation analysis and surgical planning of the craniomaxillofacial (CMF) region. Unlike most of the methods requiring segmentation of the object of interest for precise landmarking, our purpose is to perform anatomical landmarking using the inherent relation among the anatomy in the CMF region without explicitly segmenting the CMF bones. To achieve this, we propose a new deep network architecture, *relational reasoning network (RRN)*, to accurately learn the local and the global relations of the landmarks. Specifically we are interested in learning landmarks in mandible, maxilla, and nasal bones where maxilla and nasal bones are extremely difficult to segment while mandible segmentation poses unique challenges such as having the most deformities on the mandible. The proposed RRN works in an end-to-end manner, utilizing learned relations of the landmarks based on dense-block units and without the need for explicit segmentation. For a given a few landmark (spatial) locations as input, the proposed system accurately and efficiently localizes the remaining landmarks on the aforementioned bones. For a comprehensive evaluation of our proposed method, we used cone-beam computed tomography (CBCT) scans of 250 patients who have congenital deformities in their skulls. Without using any segmentation guidance, the proposed system identifies the landmark locations very accurately even when there is severe pathology or deformation in the bones. The proposed RRN has also revealed unique relationships among the landmarks that help us infer several *reasoning* about informativeness of the landmark points. Our proposed system is invariant to order of landmarks and it allowed us to discover the optimum configurations (number and location) for landmarks to be localized within the object of interest (mandible) or nearby objects (maxilla and nasal). To the best of our knowledge, this is the first of its kind algorithm finding anatomical relations of the objects using deep learning to reason the locations of the target landmarks.



(a) *Menton-Condylar Left* Relation (b) *Menton-Coronoid Left* Relation (c) *Menton-Condylar Right* Relation (d) *Menton-Coronoid Right* Relation (e) *Menton* Relations

Figure 6.2: For the input domain $L_{input} = \{Me, Cd_L, Cor_L, Cd_R, Cor_R\}$: (a)-(d) pairwise relations of landmark *Menton* a) *Menton-Condylar Left*, b) *Menton-Coronoid Left*, c) *Menton-Condylar Right*, d) *Menton-Coronoid Right*, e) combined relations of *Menton*.

6.1 Overview of the proposed method

In the CMF region, the most frequently deformed or injured bone is the lower jaw bone *mandible*, the only mobile and the most functional [9]. In our previous study [115], we developed a framework to segment mandible from CBCT scans and identify the mandibular landmarks in a fully-automated way. Herein, we approach the same problem from a significantly different perspective and focusing on anatomical landmarking without the need for explicit segmentation, and extending the learned landmarks into other bones (maxilla and nasal) apart from mandible. Overall, we seek the answers for the following important questions:

- **Q1:** Can we automatically identify all anatomical landmarks of a bone if only a subset of the landmarks are given as input? If so, what is the least effort for performing this procedure? In other words, how many landmarks are necessary and which landmarks are more informative to perform this whole procedure?
- **Q2:** Can we identify anatomical landmarks of nasal and maxilla bones if we only know locations of a few landmarks in the mandible? In other words, do relations of landmarks hold true even when they belong to different anatomical structures (manifold)?

Although modern AI algorithms have made tremendous progress solving problems in biomedical imaging, relations between objects within the data are often not modeled as separate tasks. In this study, we explore inherent relations among anatomical landmarks at the local and global levels in order to explore availability of structured data samples helping anatomical landmark localization. Inferred from the morphological integration of the CMF bones, we claim that landmarks of the same bone should carry common properties of the bone so that one landmark should give clues about the positions of the other landmarks with respect to a common reference. This reference is often chosen as segmentation of the bone to enhance information flow, but in our study we leverage this reference point from the whole segmented bone into a reference landmark point. Throughout the text, we use the following definitions:

Definition 1: A *landmark* is an anatomically distinct point, helping clinicians to make reliable measurement about a condition, diagnosis, modeling a surgical model, or even creating a treatment plan.

Definition 2: A *relation* is defined as a geometric property between landmarks. Relations between two landmarks might include the following geometric features: size, distance, shape, and other implicit structural information. In this study, we focus on pairwise relations between landmarks.

Definition 3: A *reason* is defined as an inference about relationships of the landmarks. For instance, compared to closely localized landmarks (if given as input), a few of sparsely localized landmarks can help predicting landmarks better. The reason is that sparsely localized input landmark configuration captures the anatomy of region of interest and infers better global relationships of the landmarks.

Once relationships of landmarks are learned effectively, we can use this relationship to identify the landmarks on the same or different CMF bones without the need for a precise segmentation. Towards this goal, we propose to learn relationship of anatomical landmarks in two stages as illustrated in Figure **??**. In the first stage, pairwise relations (local) of landmarks are learned (shown as function g) with a simple neural network algorithm based on dense-blocks (*DBs*). Figure 6.2 shows example pairwise relations for different pairs of mandible landmarks. There are five sparsely localized landmarks, and the figure shows how we assess the relationship per landmark. The basis/reference is Menton, in this example, hence, four pairwise relations of Menton are illustrated. Ideally, a reliable relation should consider manifold of the data pertaining to useful structural information.

In the second stage of the proposed algorithm (shown as function f in Figure **??**), we simply combine pairwise relations of landmarks (g) with another neural network setting based on *RUs*. Figure 6.2(e) illustrates all four relationships of the landmark Menton (reference) with respect to other landmarks on the mandible. In this study, we confine ourselves to manifold data only (position of the landmarks and their geometric relations) without use of appearance information because one of our aims is to avoid explicit segmentation from our system to be able to use simple reasoning networks.

6.1.0.1 Summary of our contributions

- To our knowledge, the proposed method is the first in the literature to successfully apply the spatial reasoning of the anatomical landmarks for accurate and robust landmarking using deep learning.
- Many anatomical landmarking methods, including our previous work, [115, 117, 118] use bone segmentation as a guidance for finding the location of the landmarks on the surface of the bone. The major limitation imposed by such approaches is that it may not be always possible to have an accurate segmentation. Our proposed RRN system enables accurate prediction of anatomical landmarks without employing explicit object segmentation.
- Since efficiency is a significant barrier for medical AI applications, we explore new deep learning architecture designs for a better efficacy in the system performance. For this purpose, we propose to use variational dropout [119] and targeted dropout [**?**] for faster convergence of the overall system (~ 5 times faster than baselines).
- Unlike other studies, our data set includes highly variable bone deformities along with other challenges of the CBCT scans. Hence, the proposed algorithm is considered highly-robust and identifies anatomical landmarks accurately under varying conditions (Table 6.3).

The proposed method is described in detail in Section 6.2. Next in Section 6.3, our dataset, data augmentation method, evaluation metrics, the experiments and the experimental results are presented. We discuss and conclude our study in the Section 6.4.

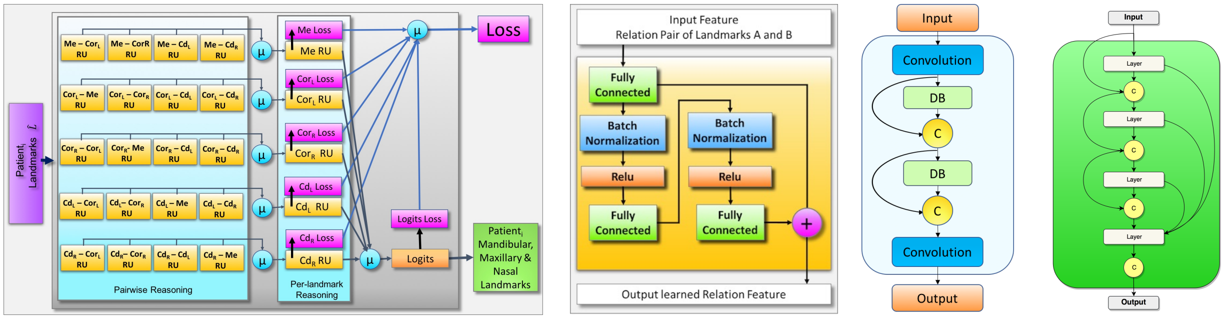
6.2 Materials and Methods

6.2.1 Relational Reasoning Architecture

Anatomical landmarking has been an active research topic for several years in the medical imaging field. The question that remains unsolved is how to build a reliable/universal relationship between landmark points for a given clinical problem. While anatomical similarities at the local and global levels are agreed to be viable solutions, thus far, features that can represent anatomical landmarks from the medical images have not achieved the desired efficacy and interpretation.

We propose a new network framework called *RRN* to learn pairwise and global relations of anatomical landmarks (o_i) through its units called *RU* (relationship unit). The relation of two landmarks are encoded as the major spatial properties of the landmarks. We define *RU* as multi-layer perceptron (*MLP*) (Figure 6.3(b)) (similar to [79]) or a network of Dense-Blocks (*DBs*) (Figure 6.3(c)) architectural unit. Our objective is to locate all anatomical landmarks by inputting a few landmarks to *RRN*, which provides reasoning inferred from the learned relationships of landmarks.

Figures 6.3(a), 6.3(b) and 6.3(c) summarize the proposed *RRN* architecture, and its *RU* sub-architectures, respectively. In the pairwise learning/reasoning stage (stage 1), 5-landmarks based system is assumed as an example network (other configurations are possible too, see experiments and results section). Sparsely-spaced landmarks (Figure **??**) and their pairwise relationships are learned in this stage (g_θ). These pairwise relationship(s) are later combined in a separate *DBs* setting in the second stage (f_ϕ). It should be noted that this combination is employed through a joint loss function and an *RU* to infer an average relation information. In other words, for each individual landmark, the combined relationship vector is assigned a secondary learning function through a single *RU* (*DBs* function f_ϕ).



(a) Relational Reasoning Network (*RRN*)

(b) MLP Relational Unit (*RU*)

(c) Dense Relational Unit (*RU*)

(d) Dense Block (*DB*) [115]

Figure 6.3: Network Architecture; a) Relational Reasoning Network for 5-input landmarks $RRN(\mathcal{L}_{input})$: L_{input}=$\{Me, Cor_L, Cor_R, Cd_L, Cd_R\}$, $\hat{L} = \{Gn, Pg, B, Id, Ans, A, Pr, Pns, Na\}$ and μ is the average operator. b) Relation Unit (RU) composed of 2 DBs, convolution and concatanation (C) units. c) Dense Block (DB) architecture composed of 4 layers and concatanation layers.

The *RU* is the core component of the *RRN* architecture and it is designed as a unit with 1 dense-block. Each dense block has a growth-rate of 4 and composed of 4 layers. Each *RU* is designed as an end-to-end fashion; hence, they are differentiable. For n landmarks in the input domain, the proposed *RRN* architecture learns $n \times (n-1)$ pairwise and n combined relations (global) with a total of n^2 *RUs*. Therefore, depending on the number of input domain landmarks, *RRN* can be either shallow or dense.

Assuming L_{input} and $\hat{L}$ indicate vectors including input and output anatomical landmarks, respectively. Then, two stages of the *RRN* of the input domain landmarks L_{input} can be defined as:

$$\begin{aligned} G_{\theta i} &= \frac{1}{(n-1)} \sum_{j=1, j\neq i}^{n} (g_\theta(o_i, o_j)), \\ RRN(L_{input}; \theta, \phi) &= \frac{1}{n} \sum_{i=1}^{n} f_{\phi_i}(G_{\theta i}). \end{aligned} \tag{6.1}$$

where $G_{\theta i}$ is the mean pairwise relation vector of the landmark o_i to every other landmark $o_{j(j \neq i)} \in L_{input}$. The functions f_ϕ and g_θ are the ***DBs*** functions with the free parameters ϕ and θ, and f_ϕ indicates a global relation (in other words, combined pairwise relations) of landmarks.

6.2.2 g_θ (pairwise relation)

For given input landmarks (L_{input}), our objective is to predict the $3D$ spatial locations of the target domain landmarks ($\in \hat{L}$) by using the $3D$ spatial locations of the input domain landmarks ($\in L_{input}$). According to the the relative locations of the input domain landmarks, we reason about the locations of the target domain landmarks. The ***RU*** function $g_\theta(o_i, o_j)$ represents the relation of two input domain landmarks o_i and o_j where $i \neq j$ (Figure 6.2). The output of $g_\theta(o_i, o_j)$ describes relative spatial context of two landmarks, defined for each pair of input domain landmarks (pairwise relation at Figure 6.3(a)). According to each input domain landmark o_i, the structure of the manifold is captured and this is represented by G_{θ_i} at Equation 6.1.

6.2.3 f_ϕ (global relation)

The mean pairwise relation $G_{\theta i}$ is calculated with respect to each input domain landmark o_i, and it is given as input to the second stage where global (combined) relation f_{ϕ_i} is learned. f_{ϕ_i} is a ***RU*** function and the output of f_{ϕ_i} is the predicted $3D$ coordinates of the target domain landmarks ($\in \hat{L}$). Each input domain landmark o_i learns and predicts the target domain landmarks by the ***RU*** function f_{ϕ_i}. The terminal prediction of the target domain landmarks is the average of individual predictions of each input domain landmark, represented by $RRN(L_{input}; \theta, \phi)$ at Equation 6.1. There are totally n^2 ***RUs***, hence n^2 ***RU*** functions, in the architecture. Note that the number of trainable parameters used for each experimental configuration are directly proportional with n^2 (Table 6.2). Since all pairwise relations are leveraged under G_{θ_i} and f_ϕ with mean operation, we can conclude that RRN is invariant to the order of landmarks in the input (i.e., permutation-invariant).

6.2.4 Loss Function

In our proposed network, the natural choice for the loss function is the mean squared error (*MSE*) because it is a differentiable distance metric measuring how well landmarks are localized/matched, and it allows output of the proposed network to be real-valued functions of the input landmarks. For n input landmarks and m target landmarks, *MSE* simply penalizes large distances between landmarks as follows:

$$Loss(\theta, \phi) = \frac{1}{n * m} \sum_{i=1}^{n} \left(\sum_{k=1}^{m} ||(f_{\phi}(G_{\theta i}))_k - o_k||^2 \right) \tag{6.2}$$

where o_k are target domain landmarks ($o_k \in \hat{L}$). The landmark localization is inherently a regression problem; therefore, *MSE* suits well to our problem. However, other loss functions can still be explored in a separate study for incremental improvements.

6.2.5 Variational Dropout

Given training input dataset $X = \{x_1, x_2, .., x_N\}$ and the corresponding output dataset $Y = \{y_1, y_2, .., y_N\}$, the goal is to learn the parameters ω such that $y = f_\omega(x)$. In the Bayesian approach, given the input and output datasets X, Y, we seek for the posterior distribution $p(\omega|X, Y)$, by which we can predict output y^* for a new input point x^* by solving the integral [**?**]:

$$p(y^*|x*, X, Y) = \int p(y^*|X*, \omega)p(\omega|X, Y)d\omega \tag{6.3}$$

In practice, this computation involves intractable integrals [119]. To obtain the posterior distributions, a Gaussian prior distribution $N(0, I)$ is placed over the network weights [**?**] which leads to a much faster convergence [119]. In our experiments, we observed a 5 times faster convergence when variational dropout is applied.

6.2.6 Targeted Dropout

Given a neural network parameterized by Θ, the goal is to find the optimal parameters W_Θ such that the loss $Loss(W_\Theta)$ is minimized. In addition, $|W_\Theta| \leq k$, i.e. only k weights of highest magnitude in the network are employed. Common solution applied is to drop the lowest $|W_\Theta| - k$ weights. In the targeted dropout, using a target rate γ and a drop out rate α, first a target set *T* is generated with the lowest weights with the target rate γ. Next, weights are stochasticity dropped out from the target set *T* with the dropout rate α [**?**]. Targeted dropout enables very fast and robust convergence of the networks in our experimental setup.

6.2.7 Features

Pairwise relations are learned through *RU* functions. Each *RU* accepts input features to be modelled as a pairwise relation. It is desirable to have such features characterizing the landmark and its interactions with other landmarks. These input features can either be learned throughout a more complicated network design, or through feature engineering. In this study, for simplicity, we define

Table 6.1: Input landmarks have the following feature(s) to be used only in stage I. $19D$ feature vector includes only structural information.

Pairwise Feature (o_A, o_B)	
3D pixel-space position of the o_A	(A_x, A_y, A_z)
Spherical coordinate of the vector from landmark Menton (o_1) to o_A	$(r_{me\to A}, \theta_{me\to A}, \phi_{me\to A})$
3D pixel-space position of the o_B	(B_x, B_y, B_z)
Spherical coordinate of the vector from landmark Menton to l_B	$(r_{me\to B}, \theta_{me\to B}, \phi_{me\to B})$
3D pixel-space position of the landmark Menton	(Me_x, Me_y, Me_z)
Spherical coordinate of the vector from o_A to o_B	$(r_{A\to B}, \theta_{A\to B}, \phi_{A\to B})$
Diagonal length of the bounding box capturing Mandible roughly, computed as the distance between the minimum and the maximum spatial locations of the input domain mandibular landmarks (L_1) in the pixel space.	d_1

a set of simple yet explainable geometric features. Since RUs model relations between two landmarks (o_A and o_B), we use $3D$ coordinates of these landmarks (both in pixel and spherical space), their relative positions with respect to each other and a well-defined landmark point (reference), and approximate size of the mandible. The mandible size is estimated as the distance between the maximum and the minimum coordinates of the input domain mandibular landmarks (Table 6.1). At final, a 19-dimensional feature vector is considered to be an input to local relationship function g in our implementation. For the reference well-defined landmark, we use *Menton (Me)* as the origin of the Mandible (See Figure 6.1(a)).

6.3 Experiments and Results

6.3.1 Data Description

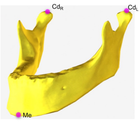
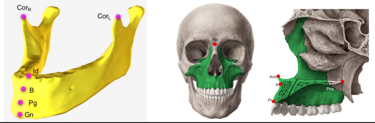
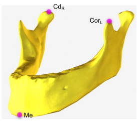
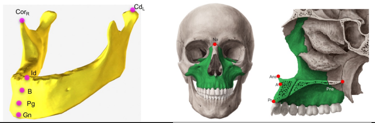
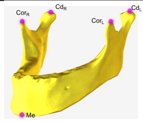
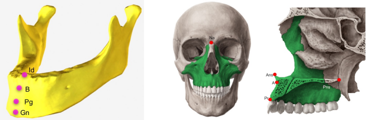
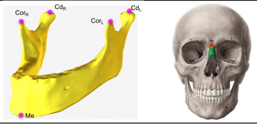
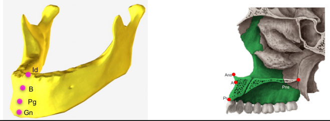
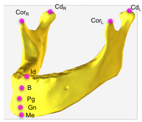
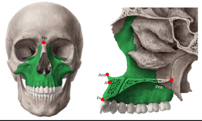
Anonymized CBCT scans of 250 patients (142 female and 108 male, mean age = 23.6 years, standard deviation = 9.34 years) were included in our analysis through an IRB-approved protocol and data sharing agreement between UCF and the NIH. The data set includes both pediatric and adult patients with craniofacial congenital birth defects, developmental growth anomalies, trauma to the CMF, and surgical interventions. CB MercuRay CBCT system (Hitachi Medical Corporation, Tokyo, Japan) was used to scan the data at 10 mA and 100 Kvp. The 12-inch field of view was required for this study to capture the entire length of the airway. The equivalent radiation dosage for each scan was approximately 300 mSv. To handle the computational cost, each patient's scan was re-sampled from $512 \times 512 \times 512$ to $256 \times 256 \times 512$. In-plane resolution of the scans were noted (in mm) either as $0.754 \times 0.754 \times 0.377$ or $0.584 \times 0.584 \times 0.292$. We also confirmed the following image-based variations in the data set: aliasing artifacts due to braces, metal alloy surgical implants (screws and plates), dental fillings, and missing bones or teeth [115].

The data was annotated independently by three expert interpreters, one from the NIH team, and two from UCF team. Among them, inter-observer agreement values were computed as circa 3 pixels. Experts used freely available $3D$ Slicer software for the annotations [115].

6.3.2 Data Augmentation

Our data set includes fully-annotated mandibular, maxillary and nasal bones' landmarks. Due to insufficiency of 250 samples for a deep-learning algorithm, we applied data-augmentation. In our study, the common usage of random scaling or rotations for data-augmentation were not found to

Table 6.2: Experimental Landmark Configurations: L_{input} and $\hat{L}$, and the number of relational units and the number of trainable parameters used in the configuration.

Configuration	$\mathbf{L_{input}}$		$\hat{\mathbf{L}}$		*#RUs*
3-Landmarks Regular	Me, Cd_L, Cd_R		Gn,Pg,B, Id, Cor_L, Cor_R, Ans,A,Pr, Pns, Na		9
3-Landmarks Cross	Me, Cd_R, Cor_L		Gn,Pg,B, Id, Cd_L, Cor_R, Ans,A,Pr, Pns, Na		9
5-landmarks	Me, Cd_L, Cd_R, Cor_L, Cor_R		Gn,Pg,B, Id,Ans,A, Pr,Pns, Na		25
6-landmarks	Me, Cd_L, Cd_R, Cor_L, Cor_R, Na		Gn,Pg,B, Id,Ans,A, Pr,Pns		36
9-landmarks	Me, Cd_L, Cd_R, Cor_L, Cor_R, Gn, Pg, B, Id		Ans,A,Pr, Pns, Na		81

be useful for new landmark data generation. Because we want to learn the specific relations of the CMF region anatomical landmarks, the scaled version of a relation of landmarks, is the same as the original relation, and the rotation would cause the loss of the available relation completely. Therefore, we instead used random interpolation in our data augmentation. Briefly, we interpolated 2 (or 3) randomly selected scans with randomly computed weight per-interpolation. We merged the relation information at different scans to a new relation. We also added a random noise to each landmark with a maximum in the range ±5 pixels, defined empirically based on the resolution of the images as well as the observed high-deformity of the bones.

6.3.3 Evaluation Methods and Metrics

We used root-mean squared error (*RMSE*) in the anatomical space (in mm) to evaluate the goodness of the landmarking algorithm. Lower *RMSE* indicates successful landmarking process. For statistical comparisons of different methods and their variants, we used P-value of 0.05 as a threshold to define significance and applied t-tests where applicable.

6.3.4 Input Landmark Configurations

In our experimental setup, there were three groups of landmarks (Figure 6.1) defined based on the bones they reside: Mandibular $L_1 = \{o_1, ..., o_9\}$, Maxillary $L_2 = \{o_{10}, ..., o_{13}\}$, and Nasal $L_3 = \{o_{14}\}$, where subscripts in o denote the specific landmark in that bone:

- $L_1 = \{Me, Gn, Pg, B, Id, Cor_L, Cor_R, Cd_L, Cd_R\}$,
- $L_2 = \{Ans, A, Pr, Pns, \}$,
- $L_3 = \{Na\}$.

In each experiment (Table 6.2), we designed a specific input set L_{input} where $L_{input} \subseteq L_1 \cup L_2$, $|L_{input}| = n$ and $1 < n <= (|L_1| + |L_2|)$. The target domain landmarks for each experiment were $\hat{L} = (L_1 \cup L_2 \cup L_3) \setminus L_{input}$ and $|\hat{L}| = m$ such that $n + m = 14$. With carefully designed input domain configurations L_{input}, and pairwise relationships of the landmarks in the input set, we seek the answers to the following questions:

- What configuration of the input landmarks can capture the manifold of bones so that other landmarks can be localized successfully?

- What is the minimum number and configuration of the input landmarks for successful identification of other landmarks?

Overall, we designed 5 different input landmark configurations called 3-landmarks regular, 3-landmarks cross, 5-landmarks, 6-landmarks and 9-landmarks (Table 6.2). Each configuration is explained in the following section.

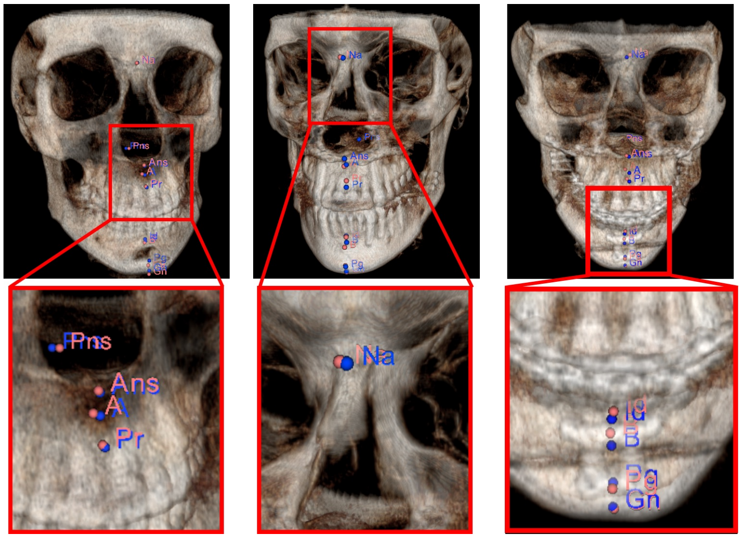
6.3.5 Experiments and Results

We ran a set of experiments to evaluate the performance of the proposed system. We summarized the experimental configurations in Table 6.2, error rates in Table 6.3 and corresponding renderings in Figure 6.4.

We designed two different *RU* architectures as Multi-Layer Perceptron (MLP) (Figure 6.3(b)) and Dense-Block (DB) (Figure 6.3(c)). DB architecture is evaluated to be more robust and fast to converge compared to the MLP architecture in our experiments. To be self-complete, we provided the MLP experimental configuration performances just for the 5-landmark experiment (See Table 6.3).

In the first experimental setup, to have an understanding of the performance of the RRN, we used the landmark grouping sparsely-spaced and closely-spaced as proposed in Neslisah et al. [115]. We named our first experimental setup as “5-landmarks” where closely-spaced, maxillary and nasal bones landmarks are predicted based on the relation of sparsely-spaced landmarks (Table 6.2). In the 5-landmarks RRN architecture, there are totally 20 *RUs*. Using the DB *RU* architecture, we trained the network for 20 epochs on 1 Nvidia Titan-XP GPU with $12GB$ memory. It takes more than 100 epochs to converge when the MLP architecture is employed.

In the second experimental setup, we explored the impact of a configuration with less number



(a) Patient-1 (b) Patient-2 (c) Patient-3

Figure 6.4: Landmark annotations using the 5-landmarks configuration: Ground truth in blue and computed landmarks in pink. a) Genioplasty/chin advancement (male 43 yo), b) Malocclusion (mandibular hyperplasia, maxillary hypoplasia) surgery (male 19 yo), c) Malocclusion (mandibular hyperplasia, maxillary hypoplasia) surgery (female 14 yo).

of input mandibular landmarks on the learning performance. Compared to the 5 sparsely-spaced input landmarks used in the first experimental configuration, herein we learned the relation of 3 landmarks, Me, Cd_L and Cd_R, and predicted the closely-spaced landmark locations (as in the 5-landmarks experiment) plus superior-anterior landmarks Cor_L and Cor_R and maxillary and nasal bones' landmark locations. The network was composed of 9 *RUs*. The training was relatively fast compared to the 5-landmarks configuration due to low number of *RUs*. We named this method as "3-Landmarks Regular". After observing statistically similar accuracy compared to the 5-landmarks method for the closely-spaced landmarks ($p - value > 0.005$), and high error rates at the superior-anterior landmarks Cor_L and Cor_R, we setup a new experiment which we named "3-Landmarks Cross".

We designed 3-landmarks cross configuration for the third experimental setup in which we used 1

Table 6.3: Landmark Localization Errors (mm). The symbol '-' means not applicable (N/A).

Method	Mandibular Landmarks						
	Cor$_\mathrm{R}$	Cor$_\mathrm{L}$	Cd$_\mathrm{L}$	Gn	Pg	B	Id
3-Landmarks Regular (Dense)	3.32 ± 0.30	3.03 ± 0.31	-	0.01 ± 0.03	0.09 ± 0.11	$\mathbf{0.60 \pm 0.15}$	0.56 ± 0.19
3-Landmarks Cross (Dense)	1.88 ± 0.24	-	1.70 ± 0.23	0.007 ± 0.03	0.10 ± 0.11	0.77 ± 0.18	0.58 ± 0.20
5-landmarks Var. Dropout (MLP)	-	-	-	0.05 ± 0.05	0.22 ± 0.13	0.91 ± 0.16	0.95 ± 0.19
5-landmarks (Dense)	-	-	-	$\mathbf{0.0002 \pm 0.03}$	0.13 ± 0.11	0.87 ± 0.16	0.78 ± 0.19
5-landmarks Var. Dropout (Dense)	-	-	-	0.0008 ± 0.02	0.07 ± 0.02	0.76 ± 0.10	0.64 ± 0.18
5-landmarks Targeted Dropout (Dense)	-	-	-	0.004 ± 0.03	$\mathbf{0.063 \pm 0.11}$	0.71 ± 0.16	0.64 ± 0.20
6-landmarks (Dense)	-	-	-	1.52 ± 0.30	0.86 ± 0.29	1.07 ± 0.25	1.24 ± 0.24
6-landmarks Var. Dropout (Dense)	-	-	-	1.04 ± 0.30	1.18 ± 0.30	0.86 ± 0.28	1.06 ± 0.24
6-landmarks Targeted Dropout(Dense)	-	-	-	1.20 ± 0.29	0.92 ± 0.28	1.09 ± 0.24	1.21 ± 0.25
9-landmarks (Dense)	-	-	-	-	-	-	-
Neslisah et al. [115]	**0.03**	**0.27**	**1.01**	0.41	1.36	0.68	**0.35**
Gupta et al. [117]	-	-	3.20	1.62	1.53	2.08	-
Method	Maxillary-Nasal Bone Landmarks						
	Ans	A	Pr	Pns		Na	
3-Landmarks Regular (Dense)	3.04 ± 0.39	3.04 ± 0.40	2.89 ± 0.40	2.04 ± 0.29		3.15 ± 0.34	
3-Landmarks Cross (Dense)	3.18 ± 0.39	3.14 ± 0.39	3.17 ± 0.38	2.61 ± 0.33		3.13 ± 0.37	
5-landmarks Var. Dropout (MLP)	3.80 ± 0.44	3.95 ± 0.48	3.06 ± 0.01	3.85 ± 0.42		3.20 ± 0.34	
5-landmarks (Dense)	3.21 ± 0.27	3.16 ± 0.41	2.92 ± 0.42	2.37 ± 0.35		2.91 ± 0.40	
5-landmarks Var. Dropout (Dense)	3.15 ± 0.21	3.07 ± 0.38	3.09 ± 0.40	2.35 ± 0.32		3.14 ± 0.36	
5-landmarks Targeted Dropout (Dense)	3.17 ± 0.38	3.09 ± 0.39	2.85 ± 0.39	2.46 ± 0.32		3.14 ± 0.40	
6-landmarks (Dense)	0.79 ± 0.23	1.65 ± 0.29	$\mathbf{1.51 \pm 0.30}$	$\mathbf{1.35 \pm 0.34}$		-	
6-landmarks Var. Dropout (Dense)	1.16 ± 0.25	$\mathbf{0.74 \pm 0.22}$	1.60 ± 0.29	1.54 ± 0.31		-	
6-landmarks Targeted Dropout (Dense)	$\mathbf{0.76 \pm 0.22}$	1.61 ± 0.28	$\mathbf{1.51 \pm 0.30}$	1.46 ± 0.36		-	
9-landmarks (Dense)	3.06 ± 0.37	3.05 ± 0.37	2.82 ± 0.35	2.42 ± 0.32		3.02 ± 0.33	
Neslisah et al. [115]	-	-	-	-		-	
Gupta et al. [117]	1.42	1.73	-	2.08		**1.17**	

superior-posterior and 1 superior-anterior landmarks on the right and left sides respectively. This network was similar to 3-landmarks regular one in terms of number of *RUs* used.

In the fourth experimental setup, we evaluated the performance of the system in learning the closely-spaced mandibular landmarks (Gn, Pg, B, Id) and the maxillary landmarks (ANS, A, Pr, PNS) using the relation information of the sparsely-spaced and the nasal-bones landmarks which is named as "6-landmarks". There are totally 36 *RUs* in this configuration.

In the fifth experimental setup, we aimed to learn the maxillary landmarks (ANS, A, Pr, PNS) and nasal bones landmark (Na) using the relation of the mandibular networks; hence, this network configuration is called "9-landmarks". The architecture was composed of 81 *RUs*. Owing to the high number of *RUs* in the architecture, the training of this network was the slowest among all the experiments performed.

We compared our results with Gupta et al. [117]. First, our results are significantly better for all landmarks except the Na landmark in spite of our highly challenging dataset which includes pediatric and adult patients with craniofacial congenital birth defects, developmental growth anomalies, trauma to the CMF, surgical intervention. The framework proposed at [117] uses an initial seed point using a 3D template registration at the inferior-anterior region where fractures are the most common. Eventually, any anatomical deformity that alters the anterior mandible may cause an error in the seed localization which can lead to a sub-optimal outcome.

We also evaluated the performance of the proposed system when variational [119] and targeted [**?**] dropouts were employed. Although statistically there was no accuracy-wise difference in the regular, variational and targeted dropout implementations, variational and targeted dropout implementations converge very fast in around epoch 20 compared to 100 of the regular dropout for the MLP architecture. Hence, for the MLP architecture, in terms of computational resources, variational and targeted dropout implementations are far more efficient for our proposed system. This is particularly important because when there are large number of *RUs*, one may focus more on the efficiency rather than accuracy. When the DB architecture is employed, we did not observe any performance improvement among different dropout implementations.

6.4 Discussions and Conclusion

We proposed an end-to-end *RRN* framework which learns the spatial dependencies between the CMF region landmarks. Based on the spatial relations of some mandibular landmarks, the rest of the landmarks on the mandible, the maxilla, and the nasal bones are identified. The proposed system is trained and tested on an augmented data set of circa $100K$ scans using 4-fold cross validation.

We hypothesized that there is an inherent relation of the CMF landmarks which can be learned using the relation reasoning architecture. In our experiments, we first evaluated this claim. We observed that 1) despite the large amount of deformities that may exist in the CMF anatomy, there is a functional relation between the CMF landmarks, and 2) *RNN* frameworks are strong enough to reveal this latent relation information. Next, we evaluated the detection performance of five different configurations of the input landmarks to find out the optimum configuration. We observed that not all landmarks are equally informative in the detection performance. Some landmark configurations are good in capturing the local information, while some have both good local and global prediction performance.

The mandibular landmarks can be classified into two groups as closely-spaced and sparsely-spaced. This grouping is useful in the reasoning architecture to grasp the informative nature of the mandibular landmarks. When we compared the 9-landmarks configuration to the 5-landmarks configuration, we observed that closely-spaced landmarks being closely-tied to the landmark Me on the mid-sagittal plane do not have additional impact on the performance of the identification of maxillary and nasal bones landmarks. In addition, in terms of the number of *RUs* and the number of trainable parameters, hence the training duration, 5-landmarks configuration is far more efficient compared to the 9-landmarks configuration. Comparing the performance of the 5-landmarks configuration to the 3 landmark configurations: 3-landmarks regular and 3-landmarks cross, first: 3-landmarks may reveal the local information closely-tied to the landmark Me for the closely-spaced landmarks, however these methods failed completely in identification of the global information: Condylar and/or Coronoid landmarks. This can be explained by the fact that 3 landmarks' configurations are not sufficient to grasp the global mandible manifold information. Interestingly, for the maxillary and nasal bone landmarks, 3-landmark configurations show statistically similar performances as the 5-landmarks configuration. This may be due to the fact that the maxillary and nasal bone landmarks are spatially in relation with the mid-sagittal plane, whereas spatially the Condylar and Coronoid landmarks are completely independent of this plane. Comparing the performances of the 5-landmarks configuration to the 6-landmarks configuration: 6-landmarks configuration perform statistically significantly better for the maxillary landmarks, in contrast, 5-landmarks configuration perform statistically significantly better for the mandibular Gn and Pg landmarks. This can be explained by the fact that, 5-landmarks configuration is sufficient to capture the local mandibular information, and extra information induced by the 6-landmarks configuration causes the precision to decrease slightly. In spite of this decrease, the 6-landmarks configuration still localizes all landmarks with an error less than 2mm. for all landmarks.

To summarize, this study focused on using the relation reasoning of the input domain CMF landmarks to predict the target domain landmarks on the mandible, maxilla, and the nasal bones. We showed that relational reasoning might be sufficient to predict the landmark locations without using the segmentation information as a guidance. As an extension study, we will design a separate deep network to learn pairwise features instead of design them ourselves. In parallel, we will incorporate appearance features from medical images to explore whether these features are superior to purely geometric features, or combined (hybrid) features can have additive values into the current research. One alternative way to pursue the research that we initiated herein will be to explore deeper and more efficient networks that can scale up the problem that we have here into a much wider platform, useful especially large number of landmarks are being explored.

CHAPTER 7: CONCLUSION

Clinicians aim to perform correct and fast interpretation of the diseases and provide treatment options for the patients. One of the most frequently used non-invasive screening method by the clinicians is medical imaging. The invaluable insight of human body provided by the medical scans is useful to the extent it can be interpreted. Two complementary parts of such an interpretation framework are medical visualization and quantitative radiology.

To visualize, volumetric medical data is projected to the viewing space in such a way that the mapping is correct, and visually meaningful and understandable frames are generated at interactive rates. Owing to the very large sizes of the current medical data scans, the clinicians usually need to work at the high-end server rooms for interpretation purposes. However, the current trend in the computing is towards the on-line applications which can be accessed from anywhere including a wide range of computing platforms. Hence, web based applications are becoming increasingly popular because browsers are supported on nearly all computing platforms. This introduces the need for a volume visualization architecture that can scale in performance by using server-based computing, yet deliver the volumes to low end devices. In the first part of this dissertation, we introduced a client-server architecture, where large data, which do not fit to neither GPU nor CPU memory is visualized at low-end mobile client devices at interactive rates.

To quantify, segmentation and anatomical landmarking are the two often used methods. In the second part of this dissertation, we presented novel quantification algorithms in the CMF region using the CT/CBCT scans. First, to segment Mandible from the CT scans, we proposed a conventional machine learning algorithm employing random forest regression, and state-machine approach. In spite of the high-accuracies obtained, traditional machine learning algorithms are limited with the nature of the hand-crafted features which may not reveal the latent information beyond human understanding. Taking into account all the limitations of the studies in the literature, we proposed a framework where deep learning algorithms are employed in the geodesic space for accurate segmentation and anatomical landmarking. We evaluated the proposed framework on a CBCT dataset with high amount of CMF deformities and imaging artifacts. As a future improvement, the Tiramisu architecture used for pixel-wise segmentation may be replaced by the Capsules [114]. In addition, a natural extension of our work will be to formulate segmentation and landmarking problem within the multi-task learning algorithm. Following this study, to further answer the question, “what if segmentation information is not available”, we introduced a deep learning framework where spatial relations of a few of the landmarks are employed to predict other landmarks in the CMF region. We found that the relations of sparsely-spaced mandibular landmarks and the Nasion landmark is sufficient for accurate detection of closely-spaced mandibular landmarks and maxillary landmarks landmark in the acceptable clinical precision. As a future extension of this study, rather than using hand-crafted features, a separate deep network may be designed to learn the pairwise features. In addition, appearance features from medical images can be explored whether these features are superior topurely geometric features, or combined (hybrid) features can have additive values into the current research.

LIST OF REFERENCES

[1] Timothy S. Newman and Hong Yi. A survey of the marching cubes algorithm. Computers & Graphics, 30:854–879, 2006.

[2] Neslisah Torosdagli. VOLREN: Web-based Interactive Real-time Volume Renderer, 2014. http://graphics.cs.ucf.edu/tools/VOLREN/.

[3] Simon Jégou, Michal Drozdzal, David Vázquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. CoRR, abs/1611.09326, 2016.

[4] Jun Zhang, Mingxia Liu, Li Wang, Si Chen, Peng Yuan, Jianfu Li, Steve Guo-Fang Shen, Zhen Tang, Ken-Chung Chen, James J. Xia, and Dinggang Shen. Joint Craniomaxillofacial Bone Segmentation and Landmark Digitization by Context-Guided Fully Convolutional Networks, pages 720–728. Springer International Publishing, Cham, 2017.

[5] Bernhard Preim and Charl P. Botha. Visual Computing for Medicine: Theory, Algorithms, and Applications. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2013.

[6] Andrey Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James Miller, Steve Pieper, and Ron Kikinis. 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network, 07 2012.

[7] Leo Grady and Gareth Funka-Lea. An energy minimization approach to the data driven editing of presegmented images/volumes. In Rasmus Larsen, Mads Nielsen, and Jon Sporring,

editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006, pages 888–895, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[8] James J. Xia, Jaime Gateno, and John F. Teichgraeber. A paradigm shift in orthognathic surgery: A special series part i. Journal of Oral and Maxillofacial Surgery, 67(10):2093–2106, 2009.

[9] A.C.V. Armond, C.C. Martins, J.C.R. Glria, E.L. Galvo, C.R.R. dos Santos, and S.G.M. Falci. Influence of third molars in mandibular fractures. part 1: mandibular anglea meta-analysis. International Journal of Oral and Maxillofacial Surgery, 46(6):716 – 729, 2017.

[10] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM.

[11] N. Max. Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics, 1(2):99–108, Jun 1995.

[12] Klaus Engel. Real-time volume graphics. A K Peters, 2006.

[13] Johanna Beyer, Markus Hadwiger, and Hanspeter Pfister. State-of-the-art in gpu-based large-scale volume visualization. Comput. Graph. Forum, 34(8):13–37, December 2015.

[14] Hanchuan Peng and Fuhui Long. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. In Nature Biotechnology, 2010.

[15] Hanchuan Peng, Alessandro Bria, Zhi Zhou, Giulio Iannello, and Fuhui Long. Extensible visualization and analysis for multidimensional images using vaa3d. Nature Protocols, 9:193–208, 2014.

[16] Hanchuan Peng, Jianyong Tang, Hang Xiao, Alessandro Bria, Jianlong Zhou, Victoria Butler, Zhi Zhou, Paloma T. Gonzalez-Bellido, Seung Wook Oh, Jichao Chen, A Mitra, Richard W. Tsien, Hongkui Zeng, Giorgio A. Ascoli, Giulio Iannello, Michael Hawrylycz, Eugene W. Myers, and Fuhui Long. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. In Nature communications, 2014.

[17] Cyril Crassin, Fabrice Neyret, Sylvain Lefebvre, and Elmar Eisemann. Gigavoxels : Ray-guided streaming for efficient and detailed voxel rendering. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), Boston, MA, Etats-Unis, feb 2009. ACM, ACM Press. to appear.

[18] Enrico Gobbetti, Fabio Marton, and Jos&#x00e9; Antonio Iglesias Guiti&#x00e1;n. A single-pass gpu ray casting framework for interactive out-of-core rendering of massive volumetric datasets. Vis. Comput., 24(7):797–806, July 2008.

[19] Barbara Maria Stafford. Body criticism: imaging the unseen in Enlightenment art and medicine. Cambridge, Mass: MIT Press, 1991.

[20] Jayaram K. Udupa and Gabor T. Herman. 3D Imaging in Medicine. CRC Press, Inc., Boca Raton, FL, USA, 1991.

[21] Qi Zhang, Roy Eagleson, and Terry M. Peters. Volume visualization: A technical overview with a focus on medical applications. Journal of Digital Imaging, 24(4):640–664, Aug 2011.

[22] W. Heiden, T. Goetze, and J. Brickmann. Fast generation of molecular surfaces from 3d data fields with an enhanced &ldquo;marching cube&rdquo; algorithm. J. Comput. Chem., 14(2):246–250, February 1993.

[23] P. J. Yim, G. B. C. Vasbinder, V. B. Ho, and P. L. Choyke. Isosurfaces as deformable models for magnetic resonance angiography. IEEE Transactions on Medical Imaging, 22(7):875–881, July 2003.

[24] Robert Stein, Alan M. Shih, M. Pauline Baker, Carl F. Cerco, and Mark R. Noel. Scientific visualization of water quality in the chesapeake bay. In Proceedings of the Conference on Visualization ’00, VIS ’00, pages 509–512, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

[25] B. A. Payne and A. W. Toga. Surface mapping brain function on 3d models. IEEE Computer Graphics and Applications, 10(5):33–41, Sept 1990.

[26] A. Wallin. Constructing isosurfaces from ct data. IEEE Computer Graphics and Applications, 11(6):28–33, Nov 1991.

[27] Hiroshi Nagahashi Takanori Nagae, Takeshi Agui. Surface construction and contour generation from volume data, 1993.

[28] M. Tory, N. Rober, T. Moller, A. Celler, and M. S. Atkins. 4d space-time techniques: a medical imaging case study. In Visualization, 2001. VIS ’01. Proceedings, pages 473–592, Oct 2001.

[29] K.S Delibasis, G.K Matsopoulos, N.A Mouravliansky, and K.S Nikita. A novel and efficient implementation of the marching cubes algorithm. Computerized Medical Imaging and Graphics, 25(4):343 – 352, 2001.

[30] P. J. Yim, G. B. C. Vasbinder, V. B. Ho, and P. L. Choyke. Isosurfaces as deformable models for magnetic resonance angiography. IEEE Transactions on Medical Imaging, 22(7):875–881, July 2003.

[31] S.L. Chan and E.O. Purisima. A new tetrahedral tesselation scheme for isosurface generation. Computers & Graphics, 22(1):83 – 90, 1998.

[32] Allen van Gelder and Jane Wilhelms. Topological considerations in isosurface generation. ACM Trans. Graph., 13(4):337–375, October 1994.

[33] D. C. Banks and S. Linton. Counting cases in marching cubes: toward a generic algorithm for producing substitopes. In IEEE Visualization, 2003. VIS 2003., pages 51–58, Oct 2003.

[34] Gunther H. Weber, Oliver Kreylos, Terry J. Ligocki, John M. Shalf, Hans Hagen, Bernd Hamann, and Kenneth I. Joy. Extraction of crack-free isosurfaces from adaptive mesh refinement data. In Gerald Farin, Bernd Hamann, and Hans Hagen, editors, Hierarchical and Geometrical Methods in Scientific Visualization, pages 19–40, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[35] C. Weigle and D. C. Banks. Extracting iso-valued features in 4-dimensional scalar fields. In IEEE Symposium on Volume Visualization (Cat. No.989EX300), pages 103–110, Oct 1998.

[36] Jane Wilhelms and Allen Van Gelder. Octrees for faster isosurface generation. ACM Trans. Graph., 11(3):201–227, July 1992.

[37] Thomas Gerstner. Fast multiresolution extraction of multiple transparent isosurfaces. In David S. Ebert, Jean M. Favre, and Ronald Peikert, editors, Data Visualization 2001, pages 35–44, Vienna, 2001. Springer Vienna.

[38] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. SIGGRAPH Comput. Graph., 22(4):65–74, June 1988.

[39] M. Levoy. Display of surfaces from volume data. IEEE Computer Graphics and Applications, 8(3):29–37, May 1988.

[40] James F. Blinn. Models of light reflection for computer synthesized pictures. In Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '77, pages 192–198, New York, NY, USA, 1977. ACM.

[41] Bui Tuong Phong. Illumination for computer generated pictures. Commun. ACM, 18(6):311–317, June 1975.

[42] Thomas Porter and Tom Duff. Compositing digital images. In Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '84, pages 253–259, New York, NY, USA, 1984. ACM.

[43] Thomas Fogal and Jens Krüger. *Tuvok*, an architecture for large scale volume rendering. In In 15th Vision, Modeling and Visualization Workshop, pages 139–146, 2010.

[44] Jason Nieh and Marc Levoy. Volume rendering on scalable shared-memory mimd architectures. In Proceedings of the 1992 Workshop on Volume Visualization, VVS '92, pages 17–24, New York, NY, USA, 1992. ACM.

[45] Chandrajit Bajaj, Insung Ihm, Gee-bum Koo, and Sanghun Park. Parallel ray casting of visible human on distributed memory architectures. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, Data Visualization '99, pages 269–276, Vienna, 1999. Springer Vienna.

[46] E Wes Bethel and Mark Howison. Multi-core and many-core shared-memory parallel raycasting volume rendering optimization and tuning. The International Journal of High Performance Computing Applications, 26(4):399–412, 2012.

[47] Gordon Kindlmann and James W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In Proceedings of the 1998 IEEE Symposium on Volume Visualization, VVS '98, pages 79–86, New York, NY, USA, 1998. ACM.

[48] M. Ruiz, A. Bardera, I. Boada, I. Viola, M. Feixas, and M. Sbert. Automatic transfer functions based on informational divergence. IEEE Transactions on Visualization and Computer Graphics, 17(12):1932–1941, Dec 2011.

[49] J. Kruger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In Proceedings of the 14th IEEE Visualization 2003 (VIS'03), VIS '03, pages 38–, Washington, DC, USA, 2003. IEEE Computer Society.

[50] M. Hadwiger, C. Berger, and H. Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In IEEE Visualization, 2003. VIS 2003., pages 301–308, Oct 2003.

[51] Enrico Gobbetti, Fabio Marton, and José Antonio Iglesias Guitián. A single-pass gpu ray casting framework for interactive out-of-core rendering of massive volumetric datasets. The Visual Computer, 24(7):797–806, Jul 2008.

[52] K. Engel. Cera-tvr: A framework for interactive high-quality teravoxel volume visualization on standard pcs. In 2011 IEEE Symposium on Large Data Analysis and Visualization, pages 123–124, Oct 2011.

[53] Aaron Knoll, Sebastian Thelen, Ingo Wald, Charles D. Hansen, Hans Hagen, and Michael E. Papka. Full-resolution interactive cpu volume rendering with coherent bvh traversal. In Proceedings of the 2011 IEEE Pacific Visualization Symposium, PACIFICVIS '11, pages 3–10, Washington, DC, USA, 2011. IEEE Computer Society.

[54] Neslisah Torosdagli, Sumanta Pattanaik, Curtis Lisle, and Yanling Liu. Web based out-of-core volume visualization in client-server architectures. In 2015 BioImage Informatics Conference, Washington, DC, USA, Oct 2015.

[55] S. Settapat, T. Achalakul, and M. Ohkura. Web-based 3d visualization and interaction of medical data using web3d. In Proceedings of SICE Annual Conference 2010, pages 2986–2991, Aug 2010.

[56] Hector Jacinto, Razmig Kéchichian, Michel Desvignes, Rémy Prost, and Sébastien Valette. A web interface for 3d visualization and interactive segmentation of medical images. In Proceedings of the 17th International Conference on 3D Web Technology, Web3D '12, pages 51–58, New York, NY, USA, 2012. ACM.

[57] Neslisah Torosdagli, Sumanta Pattanaik, Curtis Lisle, and Yanling Liu. Web-based interactive real-time volume rendering. In VIZBI - Visualizing Biological Data, Boston, MA, USA, March 2015.

[58] Dirk-Jan Kroon. Segmentation of the mandibular canal in cone-beam CT data. PhD thesis, Enschede, the Netherlands, December 2011.

[59] Shoaleh Shahidi, Ehsan Bahrampour, Elham Soltanimehr, Ali Zamani, Morteza Oshagh, Marzieh Moattari, and Alireza Mehdizadeh. The accuracy of a designed software for automated localization of craniofacial landmarks on cbct images. BMC Medical Imaging, 14(1):32, Sep 2014.

[60] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, pages 234–241. Springer International Publishing, Cham, 2015.

[61] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. 2016 Fourth International Conference on 3D Vision (3DV), pages 565–571, 2016.

[62] Baris Kayalibay, Grady Jensen, and Patrick van der Smagt. Cnn-based segmentation of medical imaging data. CoRR, abs/1701.03056, 2017.

[63] S. T. Gollmer and T. M. Buzug. Fully automatic shape constrained mandible segmentation from cone-beam ct data. In 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI), pages 1272–1275, May 2012.

[64] Ying Ji Chuang, Benjamin M. Doherty, Nagesh Adluru, Moo K. Chung, and Houri K. Vorperian. A novel registration-based semiautomatic mandible segmentation pipeline using computed tomography images to study mandibular development. Journal of Computer Assisted Tomography, Sept 2017.

[65] N. Torosdagli, D. K. Liberton, P. Verma, M. Sincan, J. Lee, S. Pattanaik, and U. Bagci. Robust and fully automated segmentation of mandible from ct scans. In 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), pages 1209–1212, April 2017.

[66] Y. Zhan, M. Dewan, M. Harder, A. Krishnan, and X. S. Zhou. Robust automatic knee mr slice positioning through redundant and hierarchical anatomy detection. IEEE Transactions on Medical Imaging, 30(12):2087–2100, Dec 2011.

[67] E. Cheng, J. Chen, J. Yang, H. Deng, Y. Wu, V. Megalooikonomou, B. Gable, and H. Ling. Automatic dent-landmark detection in 3-d cbct dental volumes. In 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 6204–6207, Aug 2011.

[68] A. Criminisi, D. Robertson, E. Konukoglu, J. Shotton, S. Pathak, S. White, and K. Siddiqui. Regression forests for efficient anatomy detection and localization in computed tomography scans. Medical Image Analysis, 17(8):1293 – 1303, 2013.

[69] Thomas Ebner, Darko Stern, Rene Donner, Horst Bischof, and Martin Urschler. Towards automatic bone age estimation from mri: Localization of 3d anatomical landmarks. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014, pages 421–428, Cham, 2014. Springer International Publishing.

[70] Yaozong Gao and Dinggang Shen. Context-aware anatomical landmark detection: Application to deformable model initialization in prostate ct images. In Guorong Wu, Daoqiang Zhang, and Luping Zhou, editors, Machine Learning in Medical Imaging, pages 165–173, Cham, 2014. Springer International Publishing.

[71] C. Chen, W. Xie, J. Franke, P.A. Grutzner, L.-P. Nolte, and G. Zheng. Automatic x-ray landmark detection and shape segmentation via data-driven joint estimation of image displacements. Medical Image Analysis, 18(3):487 – 499, 2014.

[72] C. Chen, D. Belavy, W. Yu, C. Chu, G. Armbrecht, M. Bansmann, D. Felsenberg, and G. Zheng. Localization and segmentation of 3d intervertebral discs in mr images by data driven estimation. IEEE Transactions on Medical Imaging, 34(8):1719–1729, Aug 2015.

[73] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes. Robust and accurate shape model matching using random forest regression-voting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9):1862–1874, Sept 2015.

[74] Christian Payer, Darko Štern, Horst Bischof, and Martin Urschler. Regressing heatmaps for multiple landmark localization using cnns. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, pages 230–238, Cham, 2016. Springer International Publishing.

[75] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. IEEE Transactions on Neural Networks, 20(1):61–80, Jan 2009.

[76] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks, 2015.

[77] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826, 2018.

[78] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, pages 4509–4517, USA, 2016. Curran Associates Inc.

[79] David Raposo, Adam Santoro, David Barrett, Razvan Pascanu, Timothy Lillicrap, and Peter Battaglia. Discovering objects and their relations from entangled scene representations, 2017.

[80] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. In NIPS, 2017.

[81] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

[82] J. Zhang, Y. Gao, L. Wang, Z. Tang, J. J. Xia, and D. Shen. Automatic craniomaxillofacial landmark digitization via segmentation-guided partially-joint regression forest model and

multiscale statistical features. IEEE Transactions on Biomedical Engineering, 63(9):1820–1829, Sept 2016.

[83] Khronos Group. OpenGL ES for the Web, 2007. `https://www.khronos.org/webgl/`.

[84] The X Toolkit Developers. The X Toolkit: WebGL for Scientific Visualization, 1999. `http://www.goXTK.com`.

[85] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3 data-driven documents. IEEE Transactions on Visualization and Computer Graphics, 17(12):2301–2309, December 2011.

[86] Johanna Beyer, Markus Hadwiger, and Hanspeter Pfister. A Survey of GPU-Based Large-Scale Volume Visualization. In R. Borgo, R. Maciejewski, and I. Viola, editors, EuroVis - STARs. The Eurographics Association, 2014.

[87] Hanchuan Peng. Vaa3D, 2015. `http://home.penglab.com/proj/vaa3d/home/index.html`,.

[88] Hans J. Johnson, M. McCormick, L. Ibáñez, and The Insight Software Consortium. The ITK Software Guide. Kitware, Inc., third edition, 2013. *In press*.

[89] ”What Is WebSocket?”, 2015. `http://WebSocket.org/`.

[90] Ulas Bagci, Xinjian Chen, and Jayaram K Udupa. Hierarchical scale-based multiobject recognition of 3-d anatomical structures. IEEE Transactions on Medical Imaging, 31(3):777–789, 2012.

[91] Ziyue Xu, Ulas Bagci, Brent Foster, Awais Mansoor, and Daniel J Mollura. Spatially constrained random walk approach for accurate estimation of airway wall surfaces. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 559–566. Springer, 2013.

[92] Krzysztof Chris Ciesielski, Jayaram K Udupa, Alexandre X Falcão, and Paulo AV Miranda. Fuzzy connectedness image segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis. Journal of Mathematical Imaging and Vision, 44(3):375–398, 2012.

[93] J.K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. Graphical Models and Image Processing, 58(3):246–261, 1996.

[94] Patrik F. Raudaschl, Paolo Zaffino, Gregory C. Sharp, Maria Francesca Spadea, Antong Chen, Benoit M. Dawant, Thomas Albrecht, Tobias Gass, Christoph Langguth, Marcel Lthi, Florian Jung, Oliver Knapp, Stefan Wesarg, Richard Mannion-Haworth, Mike Bowes, Annaliese Ashman, Gwenael Guillard, Alan Brett, Graham Vincent, Mauricio Orbes-Arteaga, David Crdenas-Pea, German Castellanos-Dominguez, Nava Aghdasi, Yangming Li, Angelique Berens, Kris Moe, Blake Hannaford, Rainer Schubert, and Karl D. Fritscher. Evaluation of segmentation methods on head and neck ct: Auto-segmentation challenge 2015. Medical Physics, 44(5):2020–2036, 2017.

[95] R. Mannion-Haworth, M. Bowes, A. Ashman, G. Guillard, A. Brett, and G. Vincent. Fully automatic segmentation of head and neck organs using active appearance models. 01 2016.

[96] Yaozong Gao and Dinggang Shen. Collaborative regression-based anatomical landmark detection. 60:9377–9401, 11 2015.

[97] Martin Urschler, Thomas Ebner, and Darko tern. Integrating geometric configuration and appearance information into a unified framework for anatomical landmark localization. Medical Image Analysis, 43(Supplement C):23 – 36, 2018.

[98] T. Albrecht, T. Gass, C. Langguth, and M. Lthi. Multi atlas segmentation with active shape model refinement for multi-organ segmentation in head and neck cancer radiotherapy planning. 12 2015.

[99] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):640–651, April 2017.

[100] Aliasghar Mortazi, Rashed Karim, Kawal Rhode, Jeremy Burt, and Ulas Bagci. Cardiacnet: Segmentation of left atrium and proximal pulmonary veins from mri using multi-view cnn. In Maxime Descoteaux, Lena Maier-Hein, Alfred Franz, Pierre Jannin, D. Louis Collins, and Simon Duchesne, editors, Medical Image Computing and Computer-Assisted Intervention MICCAI 2017, pages 377–385, Cham, 2017. Springer International Publishing.

[101] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. CoRR, abs/1608.06993, 2016.

[102] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time euclidean distance transform algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(5):529–533, May 1995.

[103] Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. ACM Comput. Surv., 40(1):2:1–2:44, February 2008.

[104] Prosenjit Bose, Anil Maheshwari, Chang Shu, and Stefanie Wuhrer. A survey of geodesic paths on 3d surfaces. Comput. Geom. Theory Appl., 44(9):486–498, November 2011.

[105] Manasi Datar, Ilwoo Lyu, SunHyung Kim, Joshua Cates, Martin A. Styner, and Ross Whitaker. Geodesic Distances to Landmarks for Dense Correspondence on Ensembles of Complex Shapes, pages 19–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[106] Geoff Hinton. RMSprop - Optimization algorithms. `https://www.coursera.org/learn/deep-neural-network/lecture/BhJlm/rmsprop`.

[107] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, November 1997.

[108] T. Glasmachers. Limits of End-to-End Learning. ArXiv e-prints, April 2017.

[109] S. Shalev-Shwartz, O. Shamir, and S. Shammah. Failures of Gradient-Based Deep Learning. ArXiv e-prints, March 2017.

[110] Patrik F. Raudaschl, Paolo Zaffino, Gregory C. Sharp, Maria Francesca Spadea, Antong Chen, Benoit M. Dawant, Thomas Albrecht, Tobias Gass, Christoph Langguth, Marcel Lthi, Florian Jung, Oliver Knapp, Stefan Wesarg, Richard Mannion-Haworth, Mike Bowes, Annaliese Ashman, Gwenael Guillard, Alan Brett, Graham Vincent, Mauricio Orbes-Arteaga, David Crdenas-Pea, German Castellanos-Dominguez, Nava Aghdasi, Yangming Li, Angelique Berens, Kris Moe, Blake Hannaford, Rainer Schubert, and Karl D. Fritscher. Evaluation of segmentation methods on head and neck ct: Auto-segmentation challenge 2015. Medical Physics, 44(5):2020–2036.

[111] K. Kian Ang, Qiang Zhang, David I. Rosenthal, Phuc Felix Nguyen-Tan, Eric J. Sherman, Randal S. Weber, James M. Galvin, James A. Bonner, Jonathan Harris, Adel K. El-Naggar, Maura L. Gillison, Richard C. Jordan, Andre A. Konski, Wade L. Thorstad, Andy Trotti, Jonathan J. Beitler, Adam S. Garden, William J. Spanos, Sue S. Yom, and Rita S. Axelrod. Randomized phase iii trial of concurrent accelerated radiation plus cisplatin with or without

cetuximab for stage iii to iv head and neck carcinoma: Rtog 0522. Journal of Clinical Oncology, 32(27):2940–2950, 2014. PMID: 25154822.

[112] R. Mannion-Haworth, M. Bowes, A. Ashman, G. Guillard, A. Brett, and G. Vincent. Fully automatic segmentation of head and neck organs using active appearance models. 01 2016.

[113] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. https://arxiv.org/abs/1710.05941, 2018.

[114] R. LaLonde and U. Bagci. Capsules for Object Segmentation. ArXiv e-prints, April 2018.

[115] N. Torosdagli, D. K. Liberton, P. Verma, M. Sincan, J. S. Lee, and U. Bagci. Deep geodesic learning for segmentation and anatomical landmarking. IEEE Transactions on Medical Imaging, pages 1–1, 2018.

[116] The Maxilla, by Kenhub. `https://www.kenhub.com/en/library/anatomy/the-maxilla`. Accessed: 2010-09-30.

[117] A. Gupta, O.P. Kharbanda, V. Sardana, R. Balachandran, and H.K. Sardana. A knowledge-based algorithm for automatic detection of cephalometric landmarks on cbct images. International Journal of Computer Assisted Radiology and Surgery, 10(11):1737–1752, Nov 2015.

[118] Florent Lalys, Simon Esneault, Miguel Castro, Lucas Royer, Pascal Haigron, Vincent Auffret, and Jacques Tomasi. Automatic aortic root segmentation and anatomical landmarks detection for tavi procedure planning. Minimally Invasive Therapy & Allied Technologies, 0(0):1–8, 2018. PMID: 30039720.

[119] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In Proceedings of the 28th International Conference on Neural

Information Processing Systems - Volume 2, NIPS'15, pages 2575–2583, Cambridge, MA, USA, 2015. MIT Press.